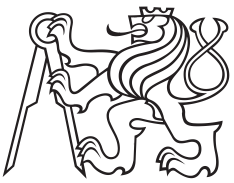
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Implementace hlasového asistenta pro nevidomé

Bc. Michal Blažek
Program: Kybernetika a robotika
Obor: Systémy a řízení

květen 2017
Vedoucí práce: Ing. Daniel Novák, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Blažek Michal**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Implementace hlasového asistenta mobilního telefonu pro nevidomé**

Pokyny pro vypracování:

1. Prostudujte problematiku ovládání mobilního telefonu hlasem s ohledem na nevidomé
2. Navrhněte systém ovládání základních funkcí telefonu hlasem
3. Implementujte systém na mobilním operačním systému Android
4. Otestujte koncept alespoň na pěti uživatelích

Seznam odborné literatury:

[1] V. Gaudissart, S. Ferreira, C. Thillou and B. Gosselin, "Sypole: A mobile assistant for the blind", 2005
[2] Shiri Azenkot, Nicole B. Lee, Exploring the use of speech input by blind people on mobile devices?, 2013
[3] L. Yujian and L. Bo, "A Normalized Levenshtein Distance Metric", 2007
[4] David M.W. Powers,Christopher C.R. Turk, ?Machine Learning of Natural Language?, 1989

Vedoucí: doc. Ing. Daniel Novák, Ph.D.

Platnost zadání: do konce letního semestru 2017/2018

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze, dne 21. 2. 2017

Poděkování / Prohlášení

Rád bych poděkoval vedoucímu své práce Ing. Danielu Novákovi, Ph.D. za výborné vedení práce. Dále bych rád poděkoval své rodině, která mě podporovala během celého studia vysoké školy, a svým přátelům.
Díky patří také všem, kteří se zúčastnili testování aplikace.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 22. 5. 2017

.....

Abstrakt / Abstract

Tato práce se zabývá návrhem hlasového ovládání mobilního telefonu pro slabozraké a nevidomé uživatele. Seznamuje čtenáře se základní problematikou a dnes dostupnými řešeními a možnostmi slabozrakých a nevidomých. Práce popisuje převod mluveného slova do textové podoby a následně zjištění významu tohoto textu. Výstupem této práce je aplikace pro mobilní telefony s operační systémem Android, která uživatelem vyslovený příkaz zpracuje a pomocí příslušné, v telefonu instalované aplikace, vykoná.

Klíčová slova: diplomová práce, Android, hlasový asistent, rozpoznání hlasu, Levenshteinův algoritmus, Jaro, Jaro-Winkler, JAVA, Speech to Text, strojové učení, hlasový výstup, slabozrací uživatelé, nevidomí uživatelé

This thesis deals with the proposal of voice control of the mobile phone for the visually impaired and the blind. It introduces readers to basic issues and today's available solutions and opportunities for the visually impaired and the blind. The paper describes the conversion of the spoken word into textual form and consequently the discovery of the meaning of this text. The output of this work is an application for Android-powered mobile phones that handles a user spoken command and performs an installed application.

Keywords: magister work, Android, voice assistant, voice recognition, Levenshtein algoritm, Jaro, Jaro-Winkler, JAVA, Speech to Text, machine learning, voice output, low vision users, blind users

Title translation: Implementation mobile phone voice assistant for blind users

Obsah /

Tabulky / Obrázky

Kapitola 1
Úvod

Přirozenou snahou člověka je si život a každou činnost v něm zjednodušovat. Tento fakt můžeme pozorovat v každém odvětví lidského zájmu od počátku věků. Čím více člověk s problémem přichází do styku, tím častěji přemýšlí o tom, jak by si danou činnost mohl zjednodušit či zkrátit čas pro ni potřebný.

Příkladem dnešní doby může být tzv. „Průmysl 4.0“, tedy snaha o vytvoření moderních robotizovaných a plně automatických řídících systémů, co nejvíce nezávislých na lidské obsluze. Dalším příkladem může být automatizace domácnosti a přenesení jejího ovládání na internet (tzv. Internet of thinks). Blízká budoucnost může mít tedy následující podobu: za tři minuty od zazvonění budíku a vstání z postele nám již kávovar připraví oblíbený šálek kávy a za dalších 10 minut se v garáži začne zahřívat automobil, který nastartuje v okamžiku, kdy do garáže vstoupíme.

Člověk se tedy snaží, aby technika, která jej obklopuje, byla co možná nejinteligentnější, vystačila si s co možná nejmenší interakcí s lidmi a postupem času se učila z jejich chování a zvyků.

Dalším znakem naší doby je snaha o co možná nejjednodušší interakci s technikou Ta by měla být zvládnutelná i bez studia nekonečných manuálů, odborného vzdělání či školení. Pro člověka je základním prvkem dorozumívání mluvený jazyk, a proto se přirozeně snažíme, aby tomuto jazyku rozuměla i technika okolo nás. Jednou z prvních vlaštovek, masivně rozšířených zařízení, schopných nějakým způsobem reagovat na přirozenou řeč člověka, jsou chytré telefony (smart phone), respektive hlasoví asistenti v nich integrovaní.

Pod pojmem hlasový asistent se skrývá aplikace, která rozumí ve větší či menší míře přirozenému jazyku člověka a dokáže na něj stejně přirozeně odpovídat či reagovat, a to tak, jak bychom čekali od člověka, který by nám dělal asistenta. Osobu, která by řešila většinu fádních každodenních věcí za nás a na nás by zbyla pouze tvůrčí činnost. Odpověď mobilního hlasového asistenta nemusí být ryze slovní, může být doplněna obrázkem či grafem nebo se může jednat přímo o akci telefonu.

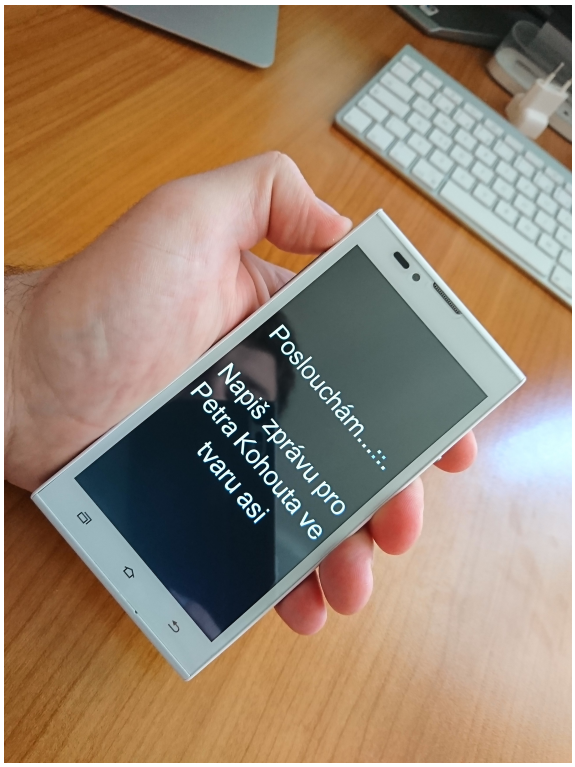
1.1 Cíl práce

Ve své práci budu vytvářet aplikaci pro mobilní telefony s operačním systémem Android. Android je operační systém založený na jádře Linuxu, který je vyvíjen firmou Google a dostupný jako open source. Podíl Androidu mezi operačními systémy chytrých telefonů je v současné době asi 80 %[1] .

Tato aplikace, jejímž účelem je ovládat klíčové funkce telefonu a případně pomoci zodpovídat i obecnější dotazy, bude sloužit hlavně pro potřeby uživatelů s vadou zraku [2] [3] [4], na což bude brán ohled při navrhování ovládání. Z tohoto důvodu bude ovládání celé aplikace ozvučené tak, aby uživateli poskytovalo zpětnou vazbu o dění na displeji telefonu. Na rozdíl od jiných mobilních asitentů nebudou informace zobrazovány

[1] `http://www.idc.com/prodserv/smartphone-os-market-share.jsp`

obrázky či grafy. Hlavní inspirace, co se týče schopností a dovedností, bude čerpána z dnes dostupných mobilních hlasových asistentů na chytrých mobilních telefonech všech operačních systémů. Kromě toho budou integrovány další funkce, přímo určené pro nevidomé a slabozraké, které budou přímo s nimi konzultovány. Aplikaci se v první fázi budu snažit vyvíjet s výhledem na to, že by měla fungovat v anglickém a českém jazyce. Reálně se ovšem budu snažit abych k řešení problémů přistupovat tak, aby bylo možné (bez zásadních zásahů do již fungující části) aplikaci rozšířit o další jazyky, případně doimplementovat další schopnosti.



Obrázek 1.1. Moje aplikace hlasového asistenta pro nevidomé

Kapitola 2
Mobilní hlasoví asistenti

Před samotným vývojem vlastního hlasového asistenta byl proveden průzkum dnes dostupných řešení pro běžné uživatele. Cílem průzkumu je ujasnit si stav a úroveň, v jaké se tato technologie aktuálně nachází. Dále se inspirovat funkcemi současných řešení, odhadnout a následně i konzultovat zdali a v jaké míře jsou důležité a implementovatelné pro potřeby nevidomých a slabozrakých.

Předem si dovolím zamyslet se na téma toho, co bychom přiblížně čekali, že by měl hlasový asistent v mobilním telefonu zvládnout pro to, aby byl uživateli opravdu přínosem. Přestože s příchodem všech sociálních sítí a všudypřítomnému internetovému připojení začíná být klasické volání a posílání zpráv přes GSM (globální systém mobilní komunikace) na ústupu, podle mého názoru jsou právě tyto dvě funkce naprosto klíčové pro hlasového asistenta a měly by jím být co nejlépe zvládnuty. Po vyslovení věty:

Zavojel Petra Novotného[1]

musí telefon poznat, že se bude volat a prohledat seznam kontaktů s cílem v něm Petra Novotného najít. Otázka k vyřešení je, co se stane když seznam kontaktů obsahuje více Petrů Novotných, případně obsahuje kontakt se jménem Petra Novotná? Tady se zjevně jedná o to, jak moc bude hlasový asistent „věřit“ tomu, že uživateli správně rozuměl a jestliže mu příliš věřit nebude, tak je i otázkou co udělat ve chvíli kdy je mezi kontakty například i Petr Novák, či Nový. Tato úloha se v podstatě týká porovnávání textových řetězců a jejich prahování.

Velmi obdobná a stejně tak stěžejní bude i reakce na příkaz:

Napiš zprávu pro Pera Nováka.

Ještě zajímavější a složitější bude ovšem úloha ve tvaru

Pošli zprávu pro Petra Nováka ve tvaru: Ahoj, Petře, jsem na cestě a budu u vás za 10 minut.

Napiš zprávu: Ahoj, Petře jsem na cestě a budu u vás za 10 minut pro Petra Nováka.

V tomto případě už bude velmi zajímavé sledovat, v jaké míře nastane situace, že telefon například nepozná kontakt, pro který je zpráva určena. Či pozorovat, zda do těla zprávy přidá nějaká slova z příkazu, která ovšem do zprávy nepatří.

Další funkcí, kterou navíc nebude problém implementovat a zároveň bude reálně nevidomým uživatelům velmi dobře sloužit, je poněkud obyčejné otevírání aplikací, které jsou v telefonu nainstalované. Pro nevidomého člověka je totiž jakékoli procházení menu telefonu složité a zdlouhavé. Uživatel musí nějakým způsobem najet na položku, následně počkat na zvukovou zpětnou vazbu, která mu řekne čeho se položka týká a teprve poté může s touto položkou pracovat nebo pokračovat na další. Samozřejmě že postupem času si člověk strukturu menu a zvláště pak často používaných položek zapamatuje a díky tomu se bude pohybovat podstatně rychleji. Otevírání aplikace prostým vyslovením jejího názvu bude pro uživatele ovšem jistě ještě jednodušší.

[1] Petr Novotný je v tomto případě absolutně fiktivní a vymyšlená osoba, sloužící pouze pro lepší demonstraci příkladů.

Obrázek 2.1. Ideové schéma aplikace

Soubor funkcí, které dříve či později potřebuje každý uživatel telefonu, je obrovský a zahrnuje všelijaká nastavení. Spoustu z nich zřejmě nemá cenu integrovat do hlasového asistenta právě z toho důvodu, že je člověk používá velmi zřídka. Zbytečně by zahlcovaly rozpoznávací mechanizmus. Naproti tomu ovšem možnost zapnout / vypnout wifi, bluethoot a podobně může být, dle mého názoru, velmi často užívaná a může uživateli reálně ušetřit čas.

Jako poslední skupinu funkcí bych uvedl různé, život zjednodušující zkratky, zlepšováky a to, co v angličtině označujeme jako „features“. Pro příklad odpovědi na otázky jako:

Jak je vysoká Eifelova věž?

Jak hrál Manchester United?

Kolik je 7x12?

2.1 Google Now

Protože svého hlasového asistenta budu vytvářet na telefonu se systémem Android, přirozeně jej budu srovnávat primárně s hlasovým asistentem Google Now, který je defaultně dostupným na této platformě. Google Now se poprvé objevil v rámci Android 4.1 „Jelly Beam“, který byl uveden v červnu roku 2012. Z počátku byl dostupný pouze uživatelům telefonů a tabletů řady Nexus, tedy těm s absolutně čistým Androidem, vyráběným přímo pod hlavičkou Googlu. Postupně se ale rozšířil a nyní je součástí všech telefonů s Androidem 4.4 a vyšším.

Za těchto necelých pět let se funkce Google Now pomalu rozvíjely. Mezi nejznámější a nejoblíbenější zlepšení určitě patří funkce oslovení telefonu. Stačí vyslovit „*OK Google*“, telefon se probudí a poslouchá váš další povel. Můžete jej tedy ovládat bez jediného dotyku. Skutečná funkčnost se ovšem razantně liší dle jazyka a při prvním spuštění je třeba nejprve třikrát vyslovit „*OK Google*“, aby se aplikace naučila vlastnosti Vašeho hlasu a fungovala lépe.

2.1.1 Volání

Nejprve se pokusíme zjistit, jak vlastně Google Now reaguje na jeden z nejtypičtějších příkazů při ovládání telefonu hlasem, tedy na příkaz k vytočení kontaktu či čísla.

V tomto směru není asistentovi Google Now co vytknout. Nepodařilo se mi ani v češtině ani v angličtině najít žádné synonymum, kterému by Google Now nerozuměl. Není tedy, naštěstí, třeba stavět větu podle nějakého pevného vzoru. Nezáleží na tom, jestli pro příkaz volání použijete sloveso „*volat*“, „*zavolat*“, „*vytočit*“.

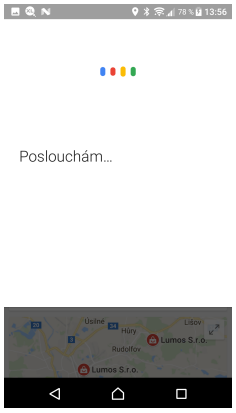
Bez problémů si poradí i s českým skloňováním nebo košatějšími větami jako:

Rád bych zavolal Petra Nováka.

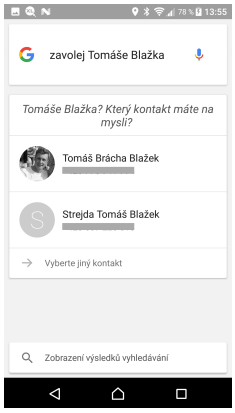
Vytoč mi, prosím, Petra Nováka.

Chtěl bych prozvonit Petra Nováka.

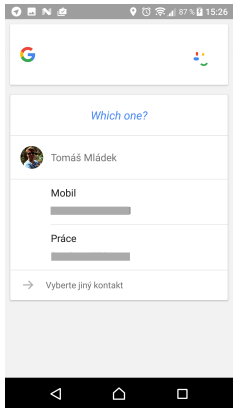
Obdobná sitace je i v angličtině, kde za klíčová slova můžeme označit „*call*“, „*dial*“ či „*ring*“. Stejně tak si poradí i s delšími větami, vedoucími k volání.



Obrázek 2.2. Google Now naslouchá



Obrázek 2.3. Více odpovídajících kontaktů



Obrázek 2.4. Více čísel u kontaktu

Jestliže si není Google Now jistý tím, který kontakt máte namysli (viz. obrázek 2.3) nebo obsahuje-li kontakt více telefonních čísel (viz. obrázek 2.4), zobrazí se tabulka, kde si můžete vybrat, a to buď dotykem na obrazovce nebo dalším dodatečným příkazem ve stylu „*ten třetí*“ atd. Tato situace nastane například, je-li více stejných kontaktů nebo jestliže nespecifikujete jméno či příjmení.

Příjemnou vychytávku objevíme, jestliže zkusíme příkaz „*zavolej bráchu*“ atp. Existují dvě možnosti, co se může stát. První, Google Now skutečně začne vytáčet kontakt vašeho sourozence a vy se maličko s hrůzou zděsíte, jak on to vlastně ví. Druhá možnost, objeví se dialog ve stylu:

Ok, rozumím, ale nevím kdo je tvůj bratr.

Poté stačí nadiktovat jeho jméno nebo jej najít mezi kontakty a příště již bude Google Now chytřejší. Takto Google Now pracuje se členy rodiny a partnery. Zeptáte-li se na doktora či právníka, tak vás žádný vyhledávací dialog nečeká.

2.1.2 Zprávy

Odesílání zpráv je možné dvěma způsoby. První a robustnější je ten, že budete postupovat obdobně jako při vytáčení kontaktů. Příkaz bude vypadat například jako:

Napiš zprávu pro Petra Nováka.

Google Now připraví zprávu s odpovídajícím kontaktem a vy doplníte text zprávy. Samozřejmě k tomuto účelu můžete opět využít možnosti text nadiktovat.

Druhý způsob, který ovšem ne vždy vede právě k tomu, co jsme chtěli a očekávali je, že zprávu vytvoříme jedním příkazem. Například:

Vytvoř zprávu pro Petra Nováka s textem: Na horách se máme krásně, počasí nám vyšlo a vůbec se nám nebude chtít domů.

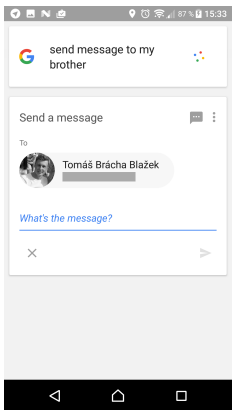
Vytvoř zprávu s textem: Na horách se máme krásně, počasí nám vyšlo a vůbec se nám nebude chtít domů pro Petra Nováka. (viz. obrázek 2.7).

Příkazy tohoto typu Google Now bez problému pochopí a skutečně správně rozklíčuje kontakt i tělo zprávy, stačí ovšem opravdu málo k tomu, aby to tak nebylo. S příkazem:

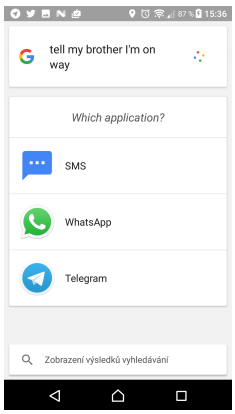
Napiš zprávu pro Petra Nováka, kde text je: Na horách se máme dobře ... atd.

již Google Now úplně dokonale nepochopí a do těla zprávy vloží text:

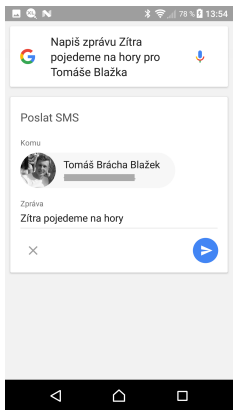
Kde text: je Na horách se máme dobře ... atd



Obrázek 2.5. Způsob psaní zprávy ve dvou krocích, nejprve komu a poté text



Obrázek 2.6. Výběr komunikačního kanálu



Obrázek 2.7. Správně nalezený příjemce i text

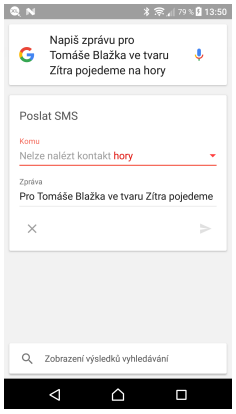
Ani anglická verze v tomto ohledu na tom není lépe a trpí absolutně totožným neduhem. Počítá tedy se skutečností, že tělo zprávy bude uvozeno souslovím „*with text*“.

2.1.3 Další možnosti

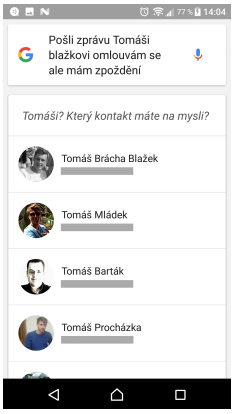
S otevíráním aplikací v telefonu nainstalovaných nemá Google Now sebemenší problém. Příkazem „*otevři*“ nebo „*zapni*“ otevře cokoli, co v telefonu máte nainstalované.

Se systémovými aplikacemi navíc umí pracovat přímo. Můžete si tedy nastavit budík na další den (viz. obrázek 2.11), vytvořit odpočet času, zapnout Wi-Fi či Bluethoot (viz. obrázek 2.13) a zřejmě ještě několik dalších věcí. Vývojáři aplikací pro android mají navíc možnost, pomocí android manifestu svou aplikaci registrovat tak, aby ji Google Now rozpoznal jako relevantní pro danou činnost a dokázal ji vyvolat a použít.

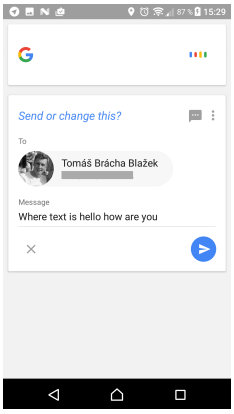
Google Now také vzorně spolupracuje s některými webovými aplikacemi Googlu, například překladačem. Prakticky do libovolného jazyka vám přeloží cokoli si řeknete a



Obrázek 2.8. Tady se opravdu nepovedlo zhola nic



Obrázek 2.9. Nepovedlo se identifikovat příjmení



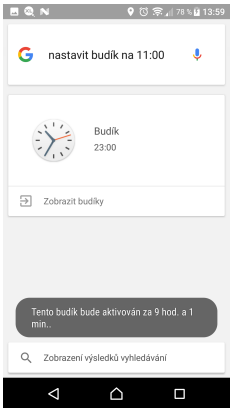
Obrázek 2.10. Příkaz „*Send message for Tomáš Blažek where text is hello how are you* “

nemusíte mít Google překladač v telefonu nainstalovaný. Stejně tak vám může ukázat aktuální hodnotu akcií firem na jednotlivých burzách či odpovědět (s využitím Wikipedie) na otázky (viz. obrázek 2.12) jako:

Jak vysoká je Eifelova věž?

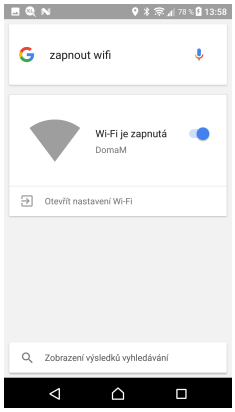
Kdy umřel Ferdinad dÉste?

Trochu zvláštně a nedeterministicky se Google Now chová v tom, jakým stylem podá odpověď. Někdy jen jednoduše ukáže výsledek na obrazovce, jindy jej i přečte.



Obrázek 2.11. Nastav budik na 11:00

Obrázek 2.12. Kdy zemřel Karel IV. ?



Obrázek 2.13. Vypni Wi-Fi

Obrázek 2.14. Počasí v Chrudimi

Zapnout Google Now je možné buď hlasovým povelem, jak již bylo řečeno, nebo dlouhým podržením home button (podle telefonu buď sowtwarového nebo hardwerového). V tomto případě ale nelze začít rovnou diktovat Vaše přání, ale je třeba ještě Google Now aktivovat do stavu, kde vás bude poslouchat, a to buď vyslovením příkazu „*OK Google*“ a nebo stisknutím ikonky mikrofonu v horní části obrazovky. Toto spouštění je tedy ve dvou krocích.

Sluší se také poznamenat co se stane, jestli že vám Google Now nerozumí a odpověď na váš dotaz nezná. Řešení je jednoduché, prostě pro váš dotaz vyhledá na internetu a zobrazí seznam odpovídajích webových stránek.

2.2 Siri

Společnost Apple, v jejíž produktech Siri najdeme, ji označuje jako inteligentní, osobní asistentku a navigátora. Siri, jejíž vývoj byl financován z organizace DARPA [1], byla poprvé uvedena v roce 2010 v Apple App Storu. Původně byla plánována jak pro iOS, tak pro Android a BlackBerry. Apple ovšem společnost, která za vývojem stála, Siri dva týdny po představení koupil, a tak byla znovu představena v roce 2011 jako exkluzivní součást iPhonu 4S.

Mezi časté výtky našinců patří lokalizce. Siri byla nejprve uvedena jen a pouze v angličtině (americké a britské). Dnes (s iOS9) je dostupná již v 27 jazycích, mezi které ovšem čeština nepatří.

Tak jako Google Now, i Siri se dá aktivovat hlasovým příkazem, v tomto případě je to „*Hey Siri*“.

2.2.1 Volání

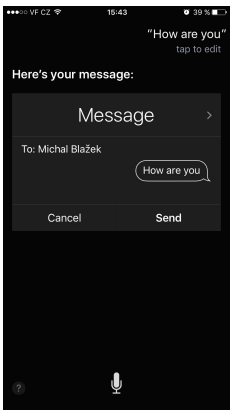
Siri stejně jako Goole now (viz. kapitola 2.1) nemá s vytáčením kontaktů a čísel zásadnější problémy. Stejně tak umí pracovat s aliasy, a tedy ani příkaz:

Call my mom.

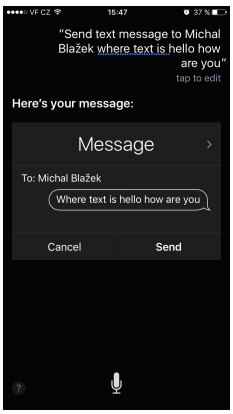
není problém (maximálně vás tedy čeká jedenkrát dialog o tom, kdo je ten který člen rodiny).

2.2.2 Zprávy

I během odesílání zpráv se Siri chová velmi podobně jako Goole now (viz. kapitola 2.1). Opravdu jsem se snažil najít podstatný rozdíl, příkaz kterému by Siri rozuměla a Google Now ne, ale v tomto hledání jsem uspěšný nebyl. Co ovšem mohu konstatovat je, že Siri je mnohem přísnější, co se týče spávné anglické výslovnosti. Google Now mi spoustu přeřeků odpustil nebo si správné slovo domyslel z kontextu věty, což ovšem u Siri úplně neplatilo a několikrát rozpoznávání hlasu vystavělo i větu zcela nesmyslnou.

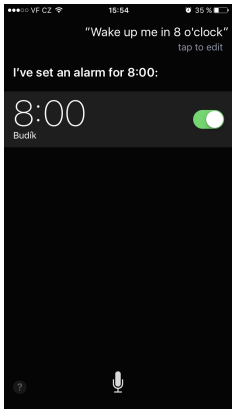


Obrázek 2.15. Příkaz „*Message for Michal Blažek with text how are you*“

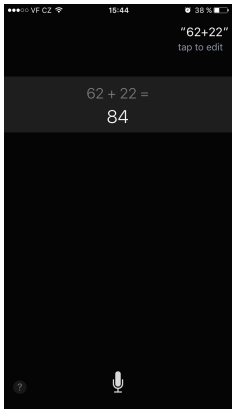


Obrázek 2.16. Příkaz „*Message for Michal Blažek where text is how are you*“

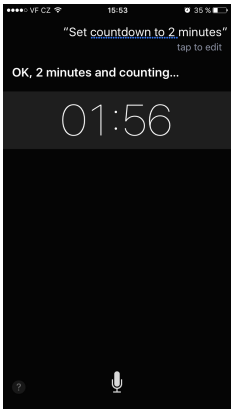
[1] `https://en.wikipedia.org/wiki/Siri`



Obrázek 2.17.
Nastavení budíku na 8:00



Obrázek 2.18.
Jednoduchý výpočet



Obrázek 2.19.
Odpočet dvou minut

2.2.3 Další možnosti

Siri se dá vzbudit buď hlasovým příkazem nebo dlouhým podržením home buttone, po kterém se objeví obrazovka Siri, která vás zpravidla pozdraví a můžete začít diktovat to, co potřebujete. Siri tedy zvládnete aktivovat z kterékoli obrazovky telefonu jedním krokem.

Opět, stejně jako u Google Now, vám i Siri dovolí nastavit několik funkcí telefonu, budík (viz. obrázek 2.17), stopky (viz. obrázek 2.19), přidat úkol do kalendáře atp. Stejně tak si můžete otevřít aplikace nainstalované v telefonu.

Siri ovšem nad Google Now vede v pochopení kontextu a v odpovědích na otázky. Můžete se ptát, jako byste mluvili s člověkem a postupně zpřesňovat dotaz. Jako třeba

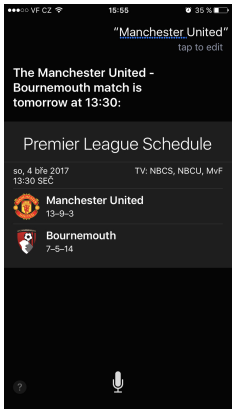
Jaký tým vede premier league?

a následně

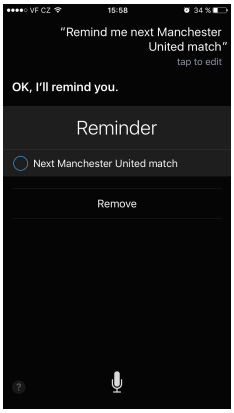
Kdo je jejich nejlepší střelec?

Kde se narodil a jak je starí?

Siri je také plná různých úsměvných funkcí nebo chcete-li „easter eggs“. Můžete se tedy Siri zeptat, jaký mobilní telefon je nejleší nebo ji přesvědčit, aby vám vyprávěla vtipy či povídky. Zpravidla jsou ale velmi krátké.



Obrázek 2.20.
Manchester United



Obrázek 2.21.
Připomenutí příštího zápasu Manchestru United

Obrázek 2.22.
„*In which city is the liberty bell*“

2.3 Cortana

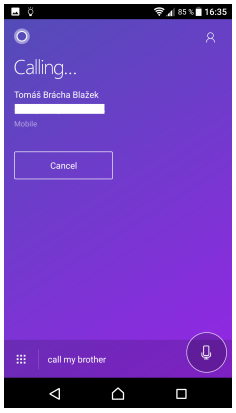
Cortana je nejmladšií asistentkou, za jejímž vývojem stojí velká technologická firma. Microsoft ji představil v dubnu roku 2014 a je součástí Windows mobile 8.1 (a vyšší) a desktopových Windows 10. Většinu dat čerpá Cortana z vyhledávače Bing (v USA je Bing populárnější než v Evropě [1]).

Hlas Cortany je odvozen od hlasu Jen Teylorové, která svůj hlas propůjčila umělé inteligenci se stejným jménem Cortana v herní serii Halo [2].

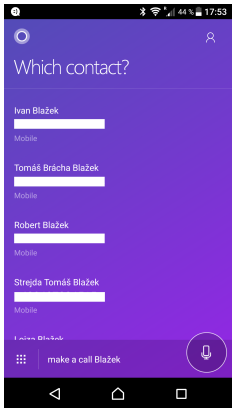
V roce 2015 Microsoft vypustil Cortanu také jako aplikaci pro Android. Do dnešního dne se ovšem„ stejně jako Siri, nenaučila česky a umí v současné době pouze osm světových jazyků.

2.3.1 Volání

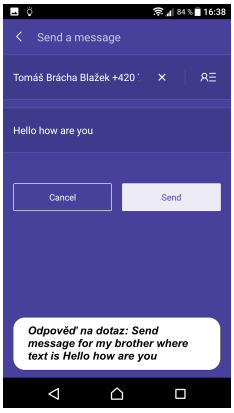
Ani Cortana se od Siri (viz. kapitola 2.2) a Google Now (viz. kapitola 2.1) nijak zásadně neliší, umí vytáčet kontakty z telefonu i telefonní čísla a stejně tak si zvládne zapamatovat rodinné příslušníky a zjednodušit tak jejich vytáčení.



Obrázek 2.23. „*Call my brother*“, protože Cortana neumí česky má problém i s českými jmény, s příkazem „*Call Tomáš Blažek*“ si neporadí



Obrázek 2.24. „*Make call to Blažek*“, tentokrát se výjmečně povedlo nadiktovat české příjmení



Obrázek 2.25. „*Send message for my brother where text is Hello how are you*“

2.3.2 Zprávy

I Cortana zvládne bez problému vytvořit SMS či email ve dvou krocích. V prvním nadiktujete, o jaký typ zprávy se bude jednat a komu ji budete posílat a ve druhém nadiktujete samotné tělo zprávy.

Cortana se snaží být chytřejší a pochopit celou zprávu najedou, ale stejně jako Siri či Google Now trpí tím, že do těla zprávy může vložit něco ze slov, co tam logicky nepatří. Podotýkám ale, že u Cortany se mi podařilo nadiktovat zprávu ve tvaru:

Message for [Kontakt] where text is [text zprávy].

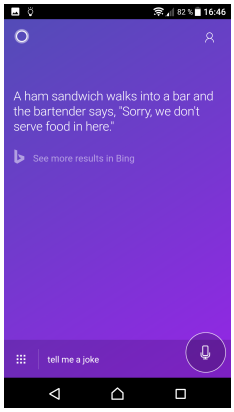
u Siri ani Google Now se mi toto nepovedlo. Skutečnost, že tímto trpí oba dva hlasoví asistenti největších technologických gigantů počítačového průmyslu dnešní doby jen dokazuje, jak složitou úlohou natural language je.

[1] `https://www.theguardian.com/technology/2015/apr/17/microsofts-bing-claims-over-20-per-cent-us-desktop-searches`
[2] `https://en.wikipedia.org/wiki/Cortana`

2.3.3 Další možnosti

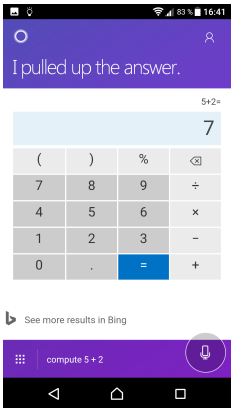
Cortana nemá žádnou funkci, kterou bych již nepopsal u Google Now (viz. kapitola 2.1) či Siri (viz. kapitola 2.2) a zároveň jí žádná z popsaných funkcí nechybí. Nedaří se jí však tak často správně odpovědět na otázku a řeší tuto skutečnost tím, že informaci vyhledá na internetu a zobrazí prvních pár stránek.

Pocitově ovšem musím říci, že Cortana zná více a lepších vtipů než Siri. Siri se začne opakovat opravdu velmi rychle, kdežto Cortana se alespoň snaží (viz. obrázek 2.27).

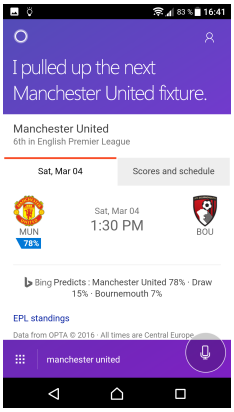


Obrázek 2.26. „*Tell me a joke*“

Obrázek 2.27. „*Tell me some story*“



Obrázek 2.28. „*Compute 5 plus 2*“



Obrázek 2.29. „*Manchester United*“

2.4 Antelli

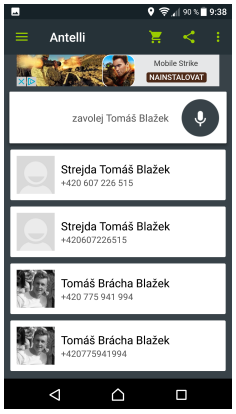
Je hlasová asistentka vytvořená vývojářem Štěpánem Šoňským, a je tedy kompletně v češtině. Hlasových asistentů je na internetu několik, jejich kvalita je ovšem velmi různorodá. Antelli sice nepřináší nic převratného a podle webu vývojáře pracuje s přesně danými příkazy a větnými celky. Tato vlastnost je poměrně omezující a vyžaduje aby uživatel možnosti a podporované tvary příkazů nastudoval. Naproti tomu má ovšem Antelli velký záběr podporovaných funkcí a umožňuje například vyhledat recept, horoskop nebo třeba program kin či cenu produktů.

2.4.1 Volání

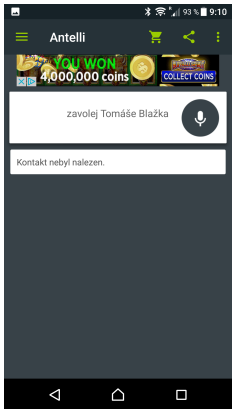
Po vzoru ostatních hlasových asistentů i Antelli umí vytáčet kontakty. Na webu vývojáře se navíc dozvídáme, na která synonyma volání reaguje. S čím si ovšem aplikace neporadí, je skloňování jmen kontaktů. Neporadí si s tím sama bez pomoci uživatele, ale za tímto účelem je možné v aplikaci přidat tzv. aliasy. Tedy další zástupná jména pro kontakt. Tímto způsobem je možné aplikaci naučit po vzoru ostatních asistentů členy rodiny a podobně.

2.4.2 Zprávy

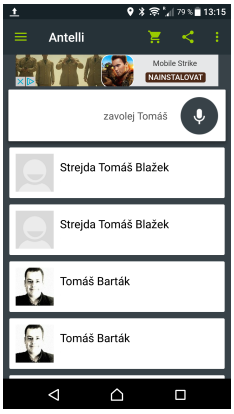
Psaní zpráv Antelli nepodporuje vůbec. Autor tak nejspíš usoudil, že oblasti, kde nedokáže dosáhnout dostatečně kvalitního výsledku, se radši vyhne. Po příkazu „*zprávy*“ Antelli ukáže zprávy ze zpravodajských webů atp.



Obrázek 2.30. „*Zavolej Tomáš Blažek*“ - v pořádku



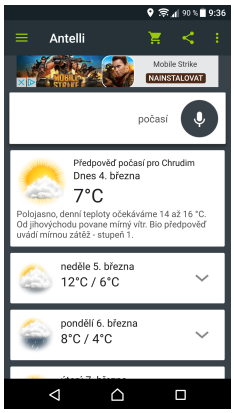
Obrázek 2.31. „*Zavolej Tomáše Blažka*“ - nic, se skloňováním si Antelli neporadí



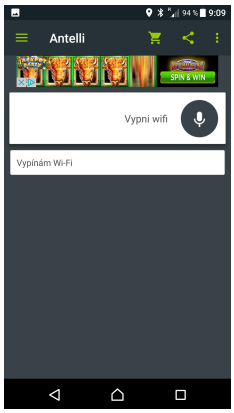
Obrázek 2.32. „*Zavolej Tomáš*“ - v pořádku, ale stačilo by změnit „Tomáš“ na „Tomáše“ a Antelli nic nenajde

2.4.3 Další možnosti

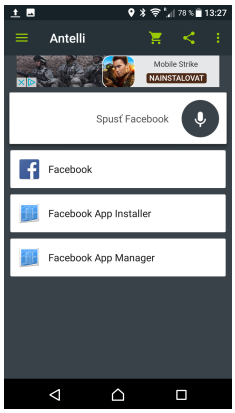
Protože má autor na webu aplikace manuál, a tedy i seznam podporovaných příkazů, není těžké zjistit, co všechno je integrované. Namátkou vyberu například recepty, horoskop, kurzy měn, svátky, programy kin či TV program. Většinou se jedná o přesměrování vyhledávání do vyhledávače jedné, někdy i více příslušných stránek a zpětné vypsání výsledků. Není to tedy pouhé otevření hledaného příkazu na Googlu nebo podobně.



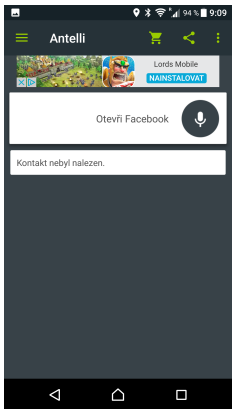
Obrázek 2.33. Počasí



Obrázek 2.34. Vypnout/ zapnout Wi-Fi



Obrázek 2.35. Otevírat aplikace v telefonu Antelli umí, ale jen příkazem „spusť“



Obrázek 2.36. Otevírat aplikace v telefonu Antelli příkazem „*otevři*“ neumí

Obrázek 2.37. Zařízení Amazon Echo (vpravo) a Google Home (vlevo) [5]

2.5 Domácí hlasoví asistenti

Velmi podobnou službu jako mobilní hlasoví asistenti zastávají i domácí hlasoví asistenti. Jedná se o zařízení, velikostí i tvarem připomínající pokojovou vázu na květiny (viz. obrázek 2.37). Zařízení se ovládá výhradně hlasovými povely. Jeho aktivace je vyvolána oslovením zařízení, obdobně jako u Google Now vyslovením „*Ok Google*“ nebo u Siri „*Hey Siri*“. Po aktivaci již zařízení poslouchá a po položení dotazu odpovídá.

Zařízení samo o sobě nedisponuje tak velkým výpočetním výkonem, aby zvládlo větu dekódovat a pochopit. Slouží tedy jako „uši“ a „ústa“ asistenta. Mozek asistenta je umísten v cloudu a zařízení je tedy plně závislé na internetovém připojení.

Domácí hlasoví asistenti výrazně mění způsob vyhledávání na internetu. Když vyhledáváme na počítači pomocí internetového prohlížeče vyhledávání většinou vrátí seřazené stránky obsahující zadaná slova, případně sousloví. Pořadí stránek se liší v závislosti například na počtu výskytu zadaných slov, nebo slov jim podobných, ale také podle návštěvnosti dané webové stránky atd. Domácí hlasový asistent, který nás vyslechne a následně nám odpoví, ovšem nemůže začíst číst hlavičky webových stránek a čekat, že si vybereme. Od domácího hlasového asistenta očekáváme lidsky podanou odpověď. Jako například:

Uživatel: *Budu dnes potřebovat deštník?*

Asistent: *Pokud půjdeš ven mezi půl jednou a půl třetí, tak ano.*

Další možností demonstrující očekávání hlasového asistenta může být následující:

Uživatel: *Kape mi kohoutek, sežeň mi opraváře, prosím!*

Asistent: *Našla jsem instalatéra Petra Nováka, sídlí 2 km odtud.*

Uživatel: *Dobře, zavolej ho!*

Asistent: *Vytáčím.*

V tomto případě již asistent nejenom hledá potřebnou informaci, ale skutečně za nás i koná a to bez toho, aniž bychom museli udělat cokoli jiného, než mluvit.

Paleta podporovaných činností je velmi široká a je otevřena vývojářům třetích stran, čímž se seznam dostupných funkcí nadále rozšiřuje. Protože jsou tato zařízení svým tvarem i designem určena pro domácí použití a jejich předpokládané umístění v domě je v obývacím pokoji, odpovídají tomuto účelu i funkce zařízení. Demonstrováno je většinou rozsvěcení či zhasínání osvětlení, změna teploty v místnostech, objednání jídla

nebo pití a podobně. Aby bylo možné ovládat právě osvětlení nebo teplotu, je třeba, abychom v domě měli podporovaný druh koncového zařízení, tedy například žárovky Philips Hue[1] nebo termostat Nest[2] a aby jejich ovládání bylo integrované v našem domácím hlasovém asistentovi.

2.5.1 Amazon Alexa

Amazon Alexa je hlasová asistentka, vyvinutá americkou firmou Amazon, která byla představena v prosinci 2014. V současné době vyrábí Amazon čtyři zařízení s přízviskem „Echo“, která Alexu obsahují. Oproti mobilním hlasovým asistentům (viz. kapitola 2) je, alespoň podle testování nezávislých novinářů, schopnost pochopení přirozené řeči na vyšší úrovni.

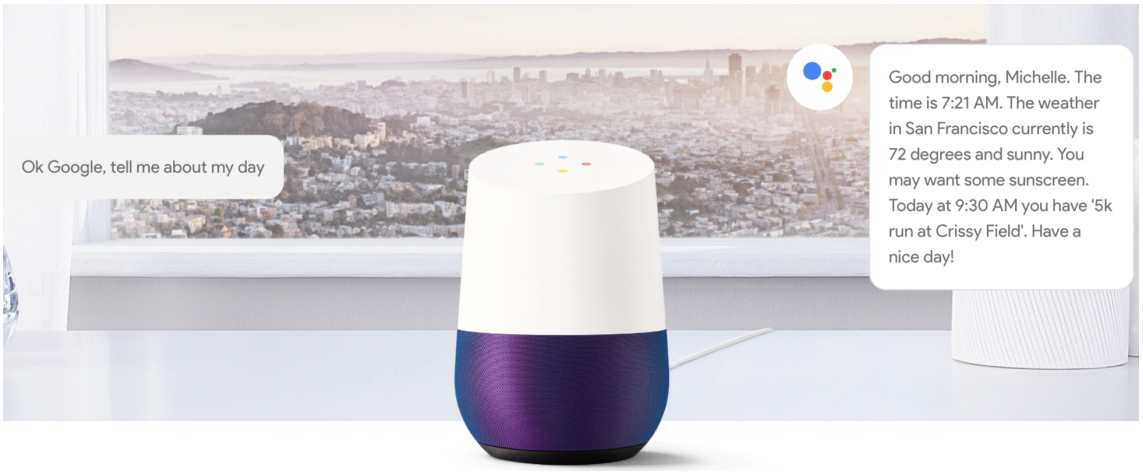
Amazon Echo tedy poslouchá a čeká na aktivační oslovení. Asistentka se aktivuje vyslovením jejího jména „Alexa“. Následně Alexa vyslechne příkaz a odešle jej do cloudu ke zpracování službou Alexa Voice Service[3]. Alexa aktuálně umí přehrávat hudbu, vytvářet poznámky, poskytovat informace o počasí, přehrávat audioknihy, streamovat podcasty poskytovat jiné další informace jako třeba o dopravě atd (viz. obrázek 2.38). Od prosince 2016 je možné navíc vaši Alexu naučit nouvou schopnost pomocí instalace „Alexa skills“. Pomocí obchodu [4] nainstalujeme skill podobně jako instalujeme aplikace do mobilních telefonů. Dostupných je více než 5000 těchto Alexa skills. Můžete díky nim Alexu změnit ve fitnes trenéra, kuchařku plnou receptů, DJ, komika, který na vás bude chrlit jeden vtip za druhým nebo třeba učitele na kytaru. V současné době, ale Alexa rozumí pouze angličtině.

Obrázek 2.38. Příklady schopností Amazon Alexa [6]

[1] https://developers.meethue.com
[2] https://nest.com
[3] https://developer.amazon.com/alexa-voice-service
[4] https://www.amazon.com/alexa-skills/b/ref=topnav_storetab_a2s?ie=UTF8&node=13727921011

2.5.2 Google Home

Google Home, stejně jako Amazon Alexa, je chytrý domácí reproduktor, který díky integrovanému Google Asistentovi odpovídá na vaše dotazy. Představen byl v květnu 2016, když vznikl evolucí z Google Now (viz. kapitola 2.1). Na rozdíl od Google Now podporuje dialogy a uchovává si dlouhodobou přestavu o kontextu po vzoru právě Alexy nebo Siri (viz. kapitola 2.2). Díky tomu je možné vést s Google Asistenten dialog podobný tomu lidskému, v kterém se náš dotaz neustále zpřesňuje na základě předchozího dotazu a odpovědi na něj.



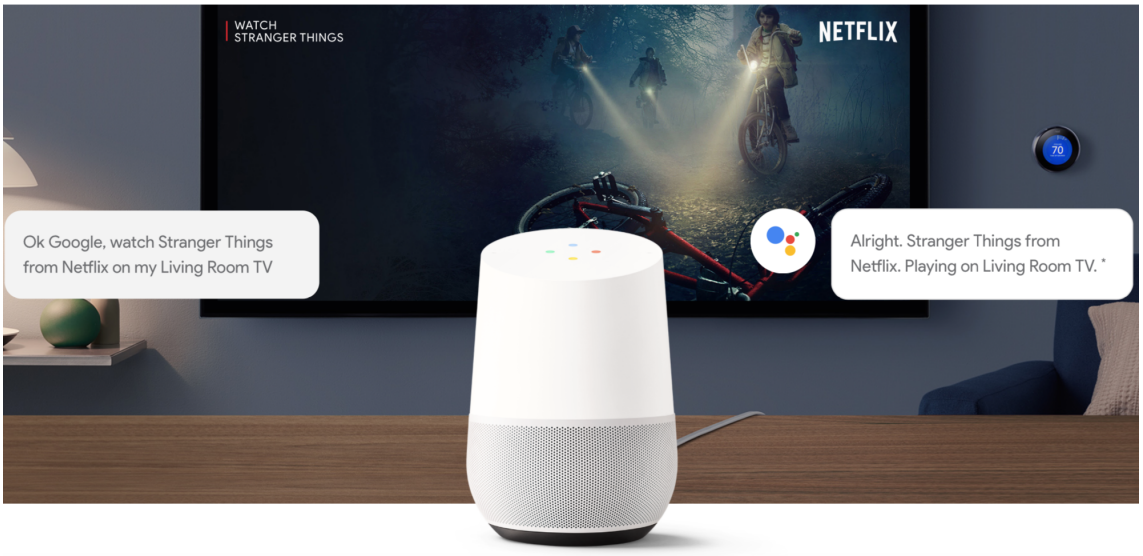
Obrázek 2.39. Dotaz na Google Asistenta - Co mě dnes čeká? [7]

Stejně jako Alexa i Google Asistent umí ovládat zařízení a aplikace třetích stran. Dokáže tedy v domě rozsvěcet a zhasínat, ovládat teplotu v místnosti nebo třeba přehrát film z Netflixu v televizi (viz. obrázek 2.40). Google Home navíc rozezná hlasy jednotlivých členů domácnosti, takže vrátí odpověď na dotazy:

Co mám na dnešek naplánováno?

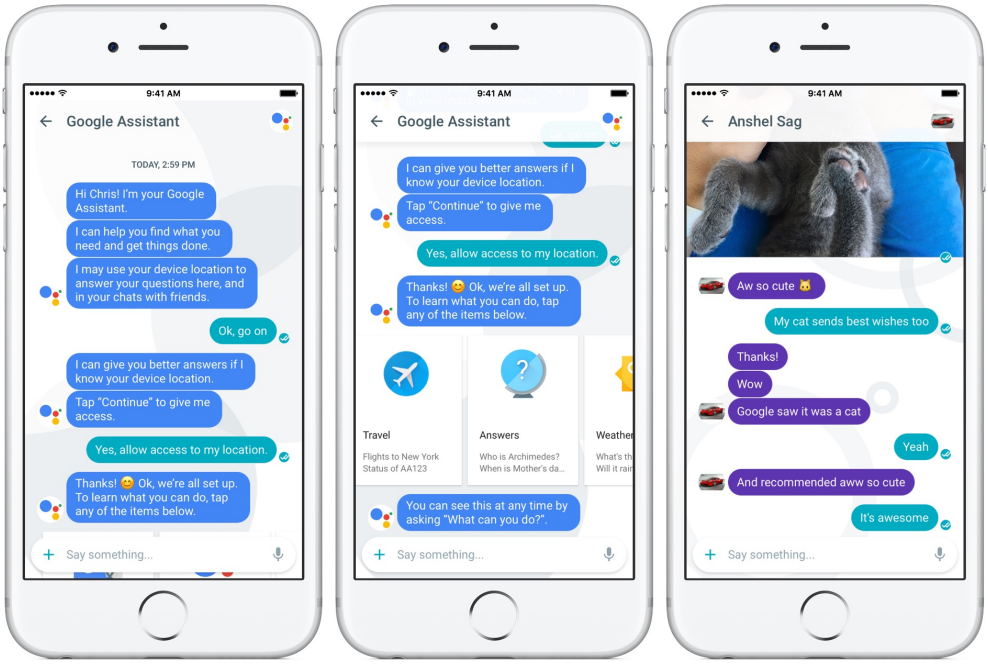
Jak dlouho pojedu dnes do práce?

na základě toho, kdo se zeptal (viz. obrázek 2.39).



Obrázek 2.40. Příkaz pro Google Asistenta - Přehraj seriál z Nteflixu [7]

Zmíněný Google Asistent integrovaný v Google Home se navíc postupně dostává i do mobilních telefonů (viz. obrázek 2.41), kde nahrazuje Google Now (viz. kapitola 2.1). Tak jako při uvedení Google Now je i nyní Google Asistent doménou telefonů vyráběných pod hlavičkou Googlu, které od té doby změnily název z Nexus na Pixel.



Obrázek 2.41. Ukázka Google Asistenta v mobilním telefonu [8]

Google navíc uvolnil SDK hlasového asistenta a tak je možné, že v blízké budoucnosti se objeví ještě další zařízení, která budou mít stejné asistenční schopnsti jako Google Home. V současné době je již možné si vlastní verzi Google Home vyrobit na základě Raspberry Pi 3.

Mezi schopnosti Google Asistenta patří tedy přehrávání hudby, vytváření událostí v kalendáři, zjišťování počasí, sportovních výsledků či zpráv a podobně. Na rozdíl od Amazonu Alexa zde není přímočará možnost rozšiřování asistentových schopností. Ty jsou přímo závislé na tom, jaká zařízení jsou registrována k vašemu google účtu.

Kapitola 3
Rozpoznávání hlasu

Naprosto klíčovou částí hlasového asistenta je jeho schopnost převodu mluveného slova do psaného textu (Speech-to-text). Dá se říci, že sebelepší algoritmus, snažící se pochopit význam vět a získat z nich co možná nejvíce relativních informací, se nemůže dostat ke správnému výsledku, nebude-li mít na vstupu správná data. Tedy pokud možno psanou verzi toho, co uživatel řekl.

3.1 Princip převodu řeči na text

Systémy převodu slova na řeč řádově starší než deset let měly typicky ještě možnost volit mezi spojitým a diskrétním rozpoznáváním řeči. Diskrétní řečí se v tomto kontextu rozumí, že uživatel vkládá mezi slova mezery výrazně delší než ty, které jsou v řeči typické. Programově je totiž mnohem jednodušší správně identifikovat slovo, je-li přesně dané, kde začíná a končí. Nicméně diktování textu diskrétně je pro uživatele nepřirozené a značně prodlužuje dobu nutnou k nadiktování věty. Téměř všechny moderní systémy dnes proto typicky pracují s řečí spojitou.

Současné programy můžeme rozdělit do dvou kategorií

3.1.1 Malý slovník / hodně uživatelů

Tyto systémy jsou typicky používané v automatizovaných telefoních automatech (IVR - interactive voice response). Jsou velmi robustní, vzhledem k množství podporovaných přízvuků a řečových vzorů. Této robustnosti je dosaženo na úkor poměrně malého množství podporovaných příkazů, jako jsou například položky v menu a podobně.

3.1.2 Velký slovník / málo uživatelů

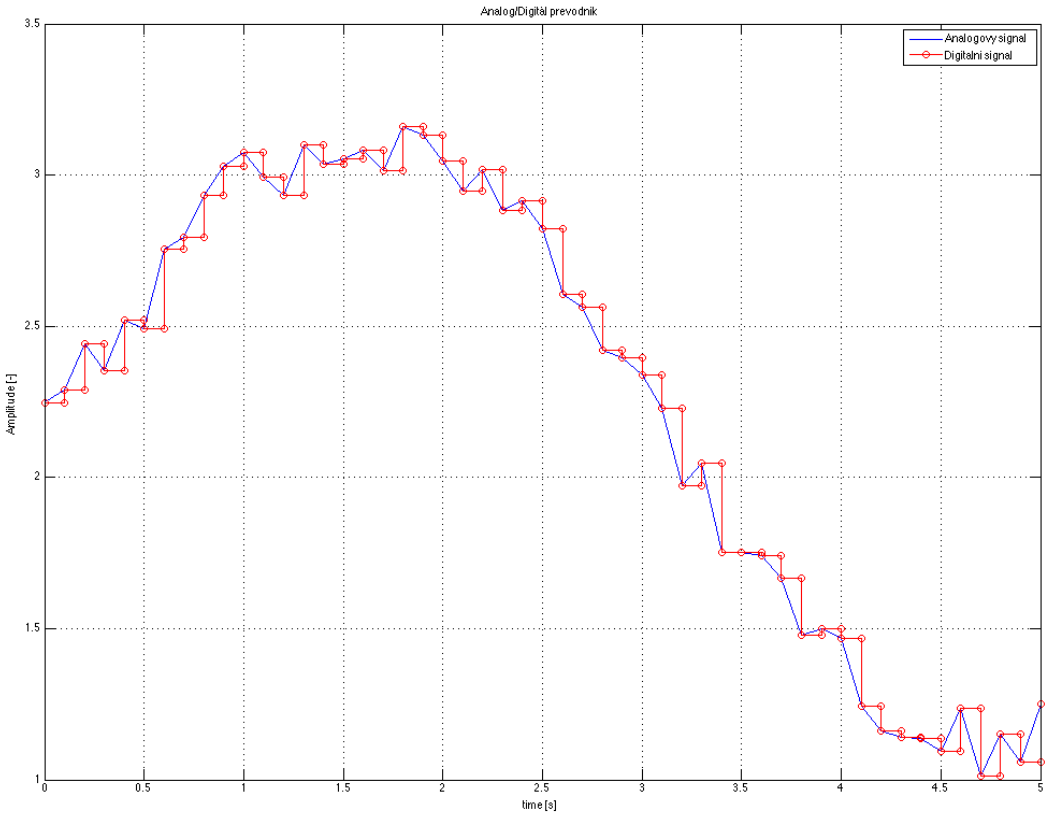
Tyto systémy se používají typicky v podnikové sféře, kde s nimi bude pracovat relativně malé procento uživatelů. I když pracují s dobrou přesností (typicky 85% a více se zkušenými uživateli), musí se trénovat a nejlépe fungují s malým okruhem primárních uživatelů. Bude-li se s tímto sytémem pokoušet pracovat uživatel, jehož přízvuk či řečový vzor je netypický pro vycvičený systém, bude míra přesnosti dramaticky klesat.

3.1.3 Převod řeči do dat

Během převodu řeči na text musí program projít několika komplexními kroky. Během mluvení člověk v podstatě vytváří vibrace ve vzduchu. Prvním krokem ve zpracování je digitalizace vstupního signálu. Na vstupu systému je tedy ADC (analog-to-digital convertor) (viz. obrázek 3.1), který transformuje vstupní analogové vlny na diskrétní data. Tato diskrétní data jsou získána odebíráním hodnoty analogové vlny v pravidelných intervalech.

V druhém kroku systém typicky filtruje digitalizovaný zvuk tak, aby se odstranil nežádoucí hluk v pozadí, někdy také, aby se zvuk rozdělil na jednotlivá frekveční pásma, která se budou následně zpracovávat samostatně. Některá frekvenční pásma se pochopitelně vypustí zcela, protože je člověk neslyší nebo v nich nemluví. V tomto kroku

také probíhá normalizace digitalizovaných dat, upravuje se hlasitost do rozsahu daného daty vzorovými. Lidé mluví různou rychlostí, a proto jsou data upravena tak, aby jejich rychlost odpovídala datům, která jsou v systému uložena jako vzorová a s kterými budou tato data následně porovnávána.



Obrázek 3.1. Princip vzorkování analogových vstupních dat

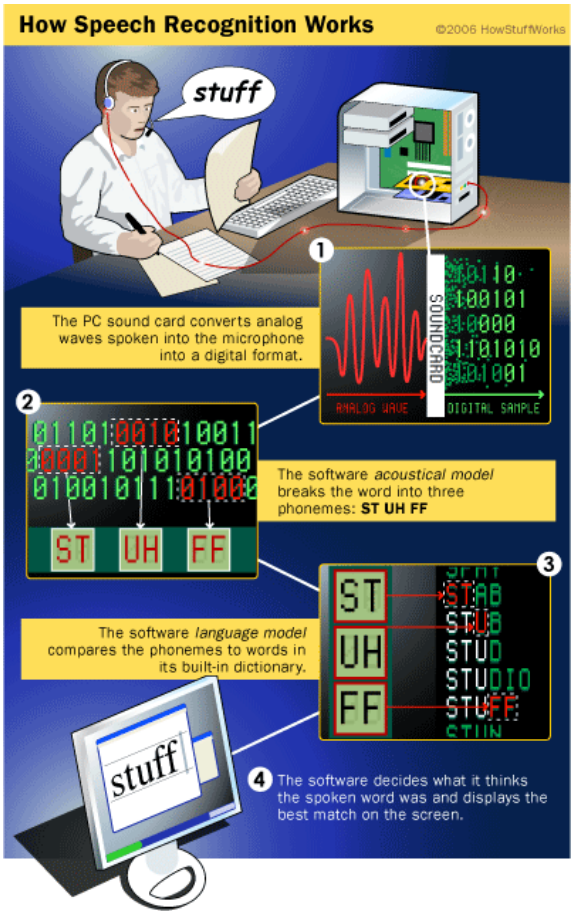
Digitalizovaný signál je následně rozdělen na velmi malé díly, krátké jenom několik setin či tisícin sekundy (v případě souhlásek jako jsou „p“ a „t“ neboli ploziva[1]). Program následně porovná tyto vzorky se známými fóny v daném jazyce. Fón je zvuk, představující určitou hlásku v daném jazyce. Složením fónů dohromady získáváme smysluplné výrazy. Anglická abecada má 26 písmen, ovšem přibližně 40 různých foném (lingvisté nejsou zcela zajedno) [9]. V češtině máme 42 písmen (včetně znaků s diakritikou a jedné spřežky), fonémů máme ovšem přibližně 39 (opět je přesný počet diskutabilní) [10].

Následující krok se může zdát snadný. Intuitivně stačí porovnat vyřčený foném s fonémy, které se v daném jazyce používají, najít ten nejpodobnější a na základě takto nalezených fonémů sestavit slovo. Realita je ovšem o poznání složitější i když princip se zasadně neliší.

Program zkoumá každý foném v kontextu fonémů okolních a hledá nejpravděpodobnější kombinace fonémů na základě komplexního statistického modelu, založeného na velké knihovně známých slov v daném jazyce. Program na tomto základě určí slovo, které uživatel nejpravděpodobněji řekl.

[1] Plozivní souhláska (ploziva, okluziva) je zvuk, který při řeči vzniká uvolněním závěru (okluze) v mluvidlech.

V dnešní době navíc pro větší přesnost dává program do kontextu nalezené slovo se slovem předposledním a případně slovy ještě staršími. Na základě zpětné vazby může zpětně přehodnotit nalezená slova v případě, že odpovídají stavbě fonémů a jsou v daném jazyce pravděpodobnější jako sousloví či věta[11].



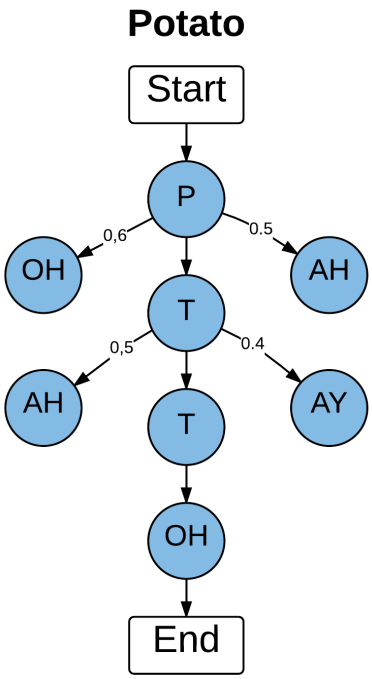
Obrázek 3.2. Proces převodu fonémů na hlásky a slova (Převzato z [12])

3.1.4 Rozpoznávání řeči a statistické modelování

Nejstarší systémy rozpoznávání řeči využívaly řadu gramatických a syntaktických pravidel. Zapadala-li slova do určitého souboru pravidel, program mohl zjistit, co tato slova znamenají. Lidský jazyk má ovšem řadu výjimek z vlastních pravidel, a to i ve chvíli, kdy člověk řádně vyslovuje a mluví spisovně. Akcenty, dialekty, případně i nálada uživatele může mít vliv na to, jak jsou slova vyslovována. Toto je také důvod, proč nejstarší systémy potřebovaly, aby řeč na vstupu byla diskrétní, tedy aby mezi slovy byly výraznější mezery.

Dnešní systémy používají rozsáhlé a výkonné statistické modely k určení nejpravděpodobnějšího výsledku. Nejčastěji používané modely v tomto směru jsou Hidden Markov Model a neuronové sítě [12]. Tyto systémy zahrnují složité matematické a statistické funkce, pomocí kterých se z informací, systému známých, snaží zjistit infomaci neznámou.

Nejčastější je Hidden Markov Model[13] (HMM). V jednodušších Markovových modelech, jako je například známý Markovův řetězec, jsou pozorovateli známy nejen výstupy modelu, ale také jeho vnitřní stavy. U HMM jsou stavy nepozorovatelné, ovšem výstup



Obrázek 3.3. Markovův model slova „potato“ s ohodnocenými hranami (Převzato z [12])

modelu pozorovatelný je. Na výstup má přitom pravděpodobnostní vliv každý vnitřní stav.

To, že je Markovův model skrytý, tedy znamená, že není známa posloupnost vnitřních stavů vedoucích k výstupu, přestože parametry modelu jsou známé. HMM nacházejí využití mimo jiné také v rozpoznávání gest a ručně psaném textu.

Na základě modelu se může program pokusit předpovědět, jaký foném přijde s největší pravděpodobností příště. Hranám grafu během tohoto procesu program přiřazuje ohodnocení na základě slovníku. Pro rozpoznávání sousloví a vět je proces ještě složitější, zejména protože program musí najít začátek a konec slova. Příkladem z angličtiny podobně znějících sousloví je „*recognize speech*“ a „*wreck a nice beach*“[12].

fonémy	r eh k ao g n ay z s p iy ch
hledané sousloví	„*recognize speech*“
fonémy	r eh k ay n ay s b iy ch
hledané sousloví	„*wreck a nice beach*“

Tabulka 3.1. Příklad prohledávání fonemů

Komplikovanost tohoto problému spočívá v tom, že slovník programu obsahuje typicky desetitisíce slov daného jazyka (nejčastěji 60000). Budeme-li hledat tři po sobě jdoucí slova, máme téměř 216 biliónů možností. Prohledat bez jakékoli další znalosti všechny možnosti, které nám v tomto případě slovník poskytuje, bude i velmi výkonnému počítači trvat dlouho.

V tomto směru se uplatňuje pomoc v podobě tréninkového programu. V trénovací fázi program dostane tisíce hodin mluveného slova společně se správným výsledkem. Na základě těchto dat se vytvoří akustické modely slov a pravděpodobnostní síť pro sousloví a věty.

Není výjimkou, že se software musí od svého koncového uživatele doučit. Typicky může být uživatel vyzván k přečtení několika vzorových vět. Systém se díky tomu může

přizpůsobit řeči uživatele (hlasitosti, rychlosti, intonaci atp.). Stejný systém se využívá i u rozpoznání frází, probouzejících mobilní asistenty Google Now a Siri (viz. kapitola 2.1) a (viz. kapitola 2.2). Uživatel také může systém naučit nová slova, typicky zkratky, odborné výrazy atp.

3.2 Přehled dostupných možností pro převod Speech-to-Text

Protože výsledná aplikace bude realizována na operačním systému Android rozebereme v tomto místě možnosti Android vývojáře pro převod mluveného slova do textové podoby.

3.2.1 Google Cloud Speech API

Google, pod jehož hlavičkou je Android vyvíjen v rámci SDK, poskytuje přístup ke svému cloudovému API na převod hlasu na text. Implementace samotná je velmi přímočará a velmi dobře zdokumentovaná.

Google Cloud Speech API rozpoznává více než 80 jazyků (včetně češtiny) a jejich variant. Zvuk dokáže zpracovat jak přímo z mikrofonu, tak z audio souboru. K rozpoznávání zvuku je použit Deep Learning neuronové sítě. Google se chlubí nepřekonatelnou přesností.

Výsledek rozpoznávání vrací Google Cloud Speech API v reálném čase a počítá přitom i s poměrně značným hlukem v okolí, vyvolaným například okolní dopravou, šumem ulice či kavárny. Funkčnost rozpoznání hlasu si může uživatel ověřit na každém telefonu s operačním systémem Android (například v Google Now (viz. kapitola 2.1)) nebo prostřednicvím implementace přímo na webu v internetovém prohlížeči. [1]

Podporováno je mnoho platforem. Toto řešení ovšem není zcela zdarma, chce-li jej vývojář použít, musí se registrovat. Prvních 60 minut rozpoznávání každý měsíc je zdarma. Každých 15 sekund nad tento limit stojí 0,006 $.

Výhodou je, že Google Speech Cloud API je součástí Android SDK. A pro vývoj na platformě Android je tedy dostupný bez jakékoli registrace, a tedy i placení. Google se patrně snaží, aby vývojáři aplikací pro jeho mobilní platformu měli rozpoznávání hlasu jednoduše k dispozici.

Je patrné, že na převodu speech-to-text se stále pracuje. Během ledna 2017 došlo k evidentní úpravě výstupu API. Do této doby totiž Google Cloud Speech API vracelo psané věty včetně interpukce, i když ne vždy zcela správně. Naproti tomu ovšem psalo čísla prakticky zásadně slovem. Od ledna 2017 se právě tyto dvě věci, a nejspíš ještě mnohé další, změnily. Například nadiktovaný čas „*osm hodin*“ se vrátí ve tvaru „*8:00*“. Stejně tak diktování, například telefonních čísel atp. bude opravdu napsáno numericky. Naproti tomu z výstupu naprosto zmizela diakritka. Působí to zvlášť úsměvně v místech, kde začíná nová věta a API správně napíše velké písmeno, ale už ne tečku na konci věty předešlé.

3.2.2 Bing speech API

I Microsoft nabízí API svého řešení převodu mluveného slova na text. Jejich řešení je v ranější fázi než řešení Googlu. Zejména podporuje mnohem méně jazyků. V současné

[1] `https://cloud.google.com/speech/`

době[1] 29 jazyků nebo plus jejich variant [2]. API se dá použít na jakémkoli zařízení připojeném k internetu. Podporovány jsou všechny (dle Microsoftu) hlavní mobilní operační systémy, tedy Android, iOS a Windows. Ale také zařízení třetích stran IoT (Internet of Thinks).

Microsoft toto řešení používá například ve své hlasové asistentce Cortana (viz. kapitola 2.3), v komunikačním nástroji Skype a v několika aplikacích pro Android.

Ke komunkaci s API může vývojář použít tři možnosti, REST API, Client nebo Server library.

Implementace do Android aplikace je samozřejmě o něco složitější než v případě Google Cloud Speech API, které je součástí SDK. Ovšem i Microsoft má v tomto směru velmi kvalitní dokumentaci, včetně vzorového projektu na GitHubu. Ovšem k tomu, aby opravdu fungovala, je třeba projít registrací ve službě Microsoft Azure a získat klíč, s nímž bude rozpoznávání samotné teprve fungovat.

Z této registrace také vyplyne cenová politika Microsoftu. Zdarma je prvních 5000 přenosů (vázaných na získaný klíč). Nad tuto kvótu Microsoft rozlišuje požadavky na rozpoznání hlasu, které jsou kratší než 15 sekund a ty, které jsou kratší než 2 minuty.[3]

Stejně jako Google i Bing speech API přepisuje vyřčená čísla jako čísla numerická. Proti Google Cloud Speech API navíc dovoluje vybrat, jestli bude poslouchat v módu, kde očekává krátkou nebo dlouho větu. Čeká-li větu krátkou, tak je-li nějakou dobu ticho, poslouchání přeruší a vrátí přepis toho, co slyšel. Naproti tomu v módu, kde očekává větu dlouhou, musí jeho činnost ukončit uživatel API. Mezitím vrací postupně jednu větu za druhou jako dílčí vysledky a v tomto módu i píše interpunkci.

3.2.3 IBM Watson Develoer Cloud - Speech to Text

V rámci svého Watson Developer Cloudu poskytuje překlad mluveného slova na text i IBM. Podporována jsou opět všechna zařízení připojená k internetu, ovšem pouze v devíti jazycích. I toto API můžete prvně otestovat ve webovém prohlížeči. V porovnání s Bing Speech nebo Google Cloud Speech je jeho výsledek výrazně horší.

Podobně jako u Microsoftu se před použitím ve vlastní aplikaci musíte prvně registrovat na portálu Bluemix, kde získáte API key nutný k fungování aplikace.

I IBM má v tomto směru podobný obchodní model jako Google a Microsoft, prvních 1000 minut měsíčně je zdarma, každá další stojí 0.02 $.

3.2.4 Kaldi

Všechny dříve popsané API byly vyíjeny pod hlavičkami velkých softwarových firem dneška. Kaldi naproti tomu je poměrně mladé open source řešení. Vzniklo během projektu „Low Development Cost, High Quality Speech Recognition for New Languages and Domains“ na Univeritě Johns Hopkins v Baltimoru v roce 2009. Po několika letech práce byl kód v květnu 2011 vypuštěn a je nadále vyvíjen a podporován.

Celý projekt je pod licencí Apache 2.0. Tato licence minimálně omezuje další použití kódu. Kód samotný je psán v jazyce C++ s důrazem na objektové pojetí, čistotu kódu a důsledně tvořenou dokumentaci.

Bohužel v současné chvíli Kaldi nepodporuje jiný jazyk než angličtinu, což vylučuje použití pro mé potřeby. Tato skutečnost je navíc pro open source speach-to-text projekty společná. Většinou se věnují právě jednomu jazyku. Případnému zájemci o použití

[1] aktuální k 21.3.2017
[2] https://www.microsoft.com/cognitive-services/en-us/speech-API/documentation/overview
[3] https://azure.microsoft.com/en-us/pricing/details/cognitive-services/speech-API/

v jiném jazyce vyjdou sice maximálně vstříc, co se týče dokumentace a aktivní uživatelské komunity, ovšem i tak je třeba odvést mnoho další práce s těžko odhadnutelným výsledkem.

3.3 Implementace vybraných Speech-to-Text API

Na základě zjištěných informací jsem se nakonec rozhodl vyzkoušet dvě API, Google Cloud Speech (viz. kapitola 3.2.1) a Bing Speech (viz. kapitola 3.2.2). Žádné objevené open source řešení pro mou potřebu vhodné není z výše popsaných důvodů a IBM Watson Developer Cloud - Speech to Text mě odradil svou nedostatečnou přesností ve webovém prohlížeči. V porovnání s vybranými řešeními od Microsoftu a Googlu byl o poznání méně přesný.

Dále bude popsán způsob implementace vybraných řešení a rozdíly mezi nimi i ve výsledku.

3.3.1 Implementace Google Cloud speech API

Protože API je v tomto případě přímo součástí SDK, je použití v Androidu velmi jednoduché [1]. Stačí inicializovat a spustit třídu a následně pouze čekat na výsledek, který se vrátí. Samozřejmě, že v tomto případě je třeba mít v manifestu povolený přístup k internetu a nahrávání audia.

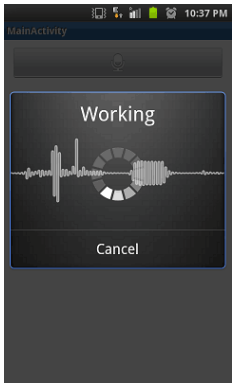
```
private static final int SPEECH_REQUEST_CODE = 0;

private void displaySpeechRecognizer() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    startActivityForResult(intent, SPEECH_REQUEST_CODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
        Intent data) {
    if (requestCode == SPEECH_REQUEST_CODE && resultCode == RESULT_OK) {
        List<String> results = data.getStringArrayListExtra(
                RecognizerIntent.EXTRA_RESULTS);
        String spokenText = results.get(0);
    }
    super.onActivityResult(requestCode, resultCode, data);
}
```

Nastavení vlastností samotného hlasového rozpoznávače probíhá skrz kontejner Bundle, který slouží k přenášení hodnot mezi Activitami. Hodnota je do kontejneru vždy uložena pod nějakým klíčem a obdobným způsobem je z něj i vyzvednuta.

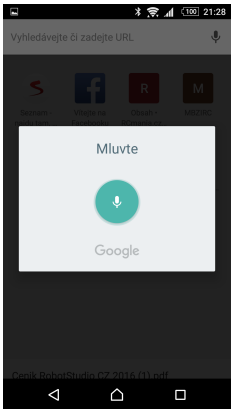
Před startem aktivity rozpoznávače hlasu se tedy do kontejneru, pomocí daných klíčových hodnot, vloží požadované parametry, jako například jazyk, v kterém bude rozpoznávač pracovat nebo čas, po kterém dojde k ukončení aktivity, je-li ticho.

Toto je opravdu nejjednodušší možnost vyvolání rozpoznávání hlasu v prostředí Androidu a k defaultní Google obrazovce (viz obrázky 3.4, 3.5, 3.6), která může být pro běžné použití naprosto dostačující, ovšem pro nevidomé uživatele vyhovující není.

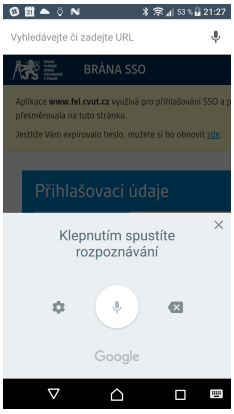
[1] https://developer.android.com/training/wearables/apps/voice.html



Obrázek 3.4. Google default recognizer dialog v Androidu 4.4



Obrázek 3.5. Google default recognizer dialog v Androidu 6.X



Obrázek 3.6. Google default recognizer dialog v Androidu 7.X

Google navíc tuto obrazovku pravidelně každoročně s novou verzí Androidu více či méně mění (viz obrázky 3.4, 3.5, 3.6), což nevidomému nepřinese žádný užitek. Z toho důvodu jsem implementoval nepatrně složitější verzi, která tento probém řeší a navíc mi dovolí celé prostředí přizpůsobit pro potřeby ovládání nevidomého uživatele.

Klíčem je vytvořit vlastní třídu implementující rozhraní `RecognitionListener`. Tuto třídu budu poté používat stejně jako standartní „Gooogle RecognitionIntent“ v předchozím případě.

Uvnitř této třídy si ovšem můžu upravit uživatelské rozhraní podle potřeb nevidomých.

Uvedu zde pouze inicializaci implementace interfacu `RecognitionListener`. Kompletní kód je součástí přílohy (viz. kapitola B).

```
    private void initSpeechRecognizer(){
        speech = SpeechRecognizer.createSpeechRecognizer(this);
        speech.setRecognitionListener(this);

        recognizerIntent = new Intent(
          RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        recognizerIntent.putExtra(
          RecognizerIntent.EXTRA_LANGUAGE,
            Locale.getDefault().getLanguage());
        recognizerIntent.putExtra(
          RecognizerIntent.EXTRA_CALLING_PACKAGE,
            this.getPackageName());
        recognizerIntent.putExtra(
          RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        recognizerIntent.putExtra(
          RecognizerIntent.EXTRA_MAX_RESULTS, 3);
        recognizerIntent.putExtra(
          RecognizerIntent.EXTRA_PARTIAL_RESULTS, true);
        recognizerIntent.putExtra(
          RecognizerIntent.EXTRA_SPEECH_INPUT_COMPLETE_
            SILENCE_LENGTH_MILLIS, 2000);
    }
```

Kompletní popis konstant je dostupný v android dokumentaci [14].

3.3.2 Implementace Bing speech API

V tomto případě je implementace o něco složitější než v případě předchozím. Celý popis je popsaný na Microsoftem odkazovaném příkladu implementace. Ve stručnosti se do Android projektu, v kterém chceme rozpoznávání Bing speech použít, přidá závislost na projektu [15], díky kterému získáme přístup přímo k Bing speech API.

Aby vše fungovalo, je třeba získat API key, kterým se naše aplikace bude na serverech Microsoftu identifikovat a bez kterého žádný výsledek nedostaneme. API key získáme na Microsoft Azure.

Vynecháme schopnost API generovat text i z audio souborů a zaměříme pozornost (stejně jako v kapitole 3.3.1) na převod mluveného slova na text. V dále uvedeném fragmentu kódu můžeme pozorovat pochopitelnou podobnost s inicializací Google Speech API.

```
this.micClient = SpeechRecognitionServiceFactory.
  createMicrophoneClientWithIntent(
    this, //odkaz na zdrojovou activitu
    this.getDefaultLocale(),
    this, //odkaz na activitu implementují
          //ISpeechRecognitionServerEvents
    this.getPrimaryKey(),
    this.getLuisAppId(),
    this.getLuisSubscriptionID());
```

Výsledek rozpoznávání hlasu následně získáme v metodě `onFinalResponseReceived(...)` ve třídě implementující rozhraní `ISpeechRecognitionServerEvents`. I v tomto případě je podobnost s implementací (viz. kapitola 3.3.1) velmi zřejmá. O významu a užití zbytku parametrů se nejvíce dozvíme v dokumentaci [16].

```
    public void onFinalResponseReceived
      (final RecognitionResult response){
        boolean isFinalDicationMessage = this.getMode() ==
          SpeechRecognitionMode.LongDictation &&
                (response.RecognitionStatus ==
                  RecognitionStatus.EndOfDictation ||
                        response.RecognitionStatus ==
                          RecognitionStatus.DictationEndSilenceTimeout);
        if (null != this.micClient) {
            this.micClient.endMicAndRecognition();
        }

        if (!isFinalDicationMessage) {
            this.WriteLine("********* Final n-BEST Results *********");
            for (int i = 0; i < response.Results.length; i++) {
                this.WriteLine("[" + i + "]" + " Confidence=" +
                 response.Results[i].Confidence +
                " Text=\"" + response.Results[i].DisplayText + "\"");
            }
        }
    }
```

Microsoft Bing speech nemá v prostředí Androidu žádné unifikované a jednotné GUI, proto není možné řešit otázku, zdali je ovládání vhodné pro nevidomé či nikoli a je třeba jej implementovat.

3.4 Konečný výběr služby na rozpoznávání hlasu

Po prozkoumání dostupných řešení bylo rozhodnuto použít pro převod mluveného slova na text službu Google Cloud Speech API (viz. kapitola 3.2.1). Toto řešení vyčnívá nad ostatními zejména počtem podporovaných jazyků, mezi kterými je i čeština, a jednoduchostí implementace. Výhodou je také, že při vysoké kvalitě výstupu není na platformě Android nijak omezený nebo zpoplatněný počet použití API.

Pro úplnost je uvedena tabulka se subjektivním hodnocením kvality převodu testovaných speech-to-text API, nejčastějšími nalezenými chybami rozhraní a dalšími informacemi, získanými během testování.

Speech API	Kvalita[%]	Nejčastější problémy	Počet podporovaných jazyků
Google cloud speech API	90	nepodporuje interpunkci, několik formátů času	80+
Bing speech API	80	několik formátů času	29
IBM Watson develoer cloud	65		9
Kaldi	65		1 (možno rozšířit)

Tabulka 3.2. Tabulka hodnocení speech-to-text API

Kapitola 4
Pochopení vyřčeného příkazu

Jakmile získáme textovou podobu, uživatelem vyřčeného příkazu, je nutné zjistit, jaké oblasti se tento příkaz týká. Pochopit tedy, co měl uživatel na mysli a příslušnou akci vykonat. K tomuto je možné použít dva přístupy [17] [18].

Prvním je rozklíčovat příkaz pomocí klíčových slov, větných sekvencí a podobně [19]. Kromě slovníku klíčových slov v tomto směru může být užitečná i informace o větných členech či slovních druzích [20] [21] [22].

Druhým způsobem může být na základě množiny vzorových, trénovacích, dat vycvičit neuronovou síť [1] [22] [23], která se díky zpětné vazbě bude neustále zlepšovat a zpřesňovat. Tento přístup ovšem vyžaduje dostatek správně ohodnocených dat, která, bohužel, v této fázi nemáme k dispozici, a proto budeme pokračovat prvním přístupem. Díky němu v budoucnu získáme dostatek správně ohodnocených dat, na jejichž základě bude možné neuronovou síť vycvičit.

4.1 Porovnávání textových řetězců

V SDK Javy není dostupný žádný jiný způsob porovnávání textových řetězců[1] než metody `eaqual()` a `contain()`. Metoda `eaqual()` ovšem porovnává řetězce `A` a `B` jako celky. Metoda `contain()` v prohledávaném řetězci `A` hledá posloupnost znaků `B`, tedy pokud `A` neobsahuje přesně posloupnost `B`, tak není shoda nalezena.

K nalezení podobnosti textových řetězců je třeba algoritmus, který číselně ohodnotí podobnost dvou stringů. Neboli vhodnou metrikou změří vzdálenost mezi nimi.

4.2 Editační vzdálenosti

Editační vzdálenosti je způsob porovnání dvou textových řetězců a označuje počet změn, nutných k transformaci jednoho stringu na druhý. Jako změna se počítá vkládání, odstranění nebo záměna znaků. Pojem editační vzdálenost „Edit distance“ [17] [24] [25] [26] často označuje právě Levenshteinovu vzdálenost (viz. kapitola 4.2.1), ve skutečnosti editační vzdálenost označuje skupinu metrik, sloužící k měření vzdálenosti mezi dvěma stringy. Jako například:

- Levenshteinova vzdálenost (viz. kapitola 4.2.1)
- Hammingova vzdálenost (viz. kapitola 4.2.2)
- Damerau–Levenshteinova vzdálenost (viz. kapitola 4.2.3)
- Jaro vzdálenost (viz. kapitola 4.2.4)
- Jaro-Winkler vzdálenost (viz. kapitola 4.2.5)

a další [27].

[1] v programování zdomácněl pro textové řetězce výraz stringy

4.2.1 Levenshteinova vzdálenost

Levenshteinova vzdálenost [28] [29] je metrika používaná k určení vzdálenosti mezi dvěma sekvencemi. Mějme dvě sekvence znaků `A` a `B`. Vzdálenost mezi dvěma sekvencemi je určena jako minimum množství změn v sekvenci `A`, díky kterému ze skvence `A` vytvoříme přesnou kopii sekvence `B`. Jako změna se počítá vkládání, odstranění nebo záměna znaků.

Vztah pro výpočet Levenshtainovy vzdálenosti pro dva textové řetězce `A` a `B` (s délkami $\|a\|$ a $\|b\|$) je dán vztahem $lev_{A,B}(\|a\|, \|b\|)$. S využitím rekurze můžeme funkci pro výpočet Levenshteinovy vzdálenosti definovat následovně:

$$lev_{A,B}(i,j) = \begin{cases} \max(i,j), & když \min(i,j) = 0; \\ \min \begin{cases} lev_{A,B}(i-1,j)+1 \\ lev_{A,B}(i,j-1)+1 \\ lev_{A,B}(i-1,j-1)+1_{A_i \neq, B_j} \end{cases}, & jindy \end{cases}$$

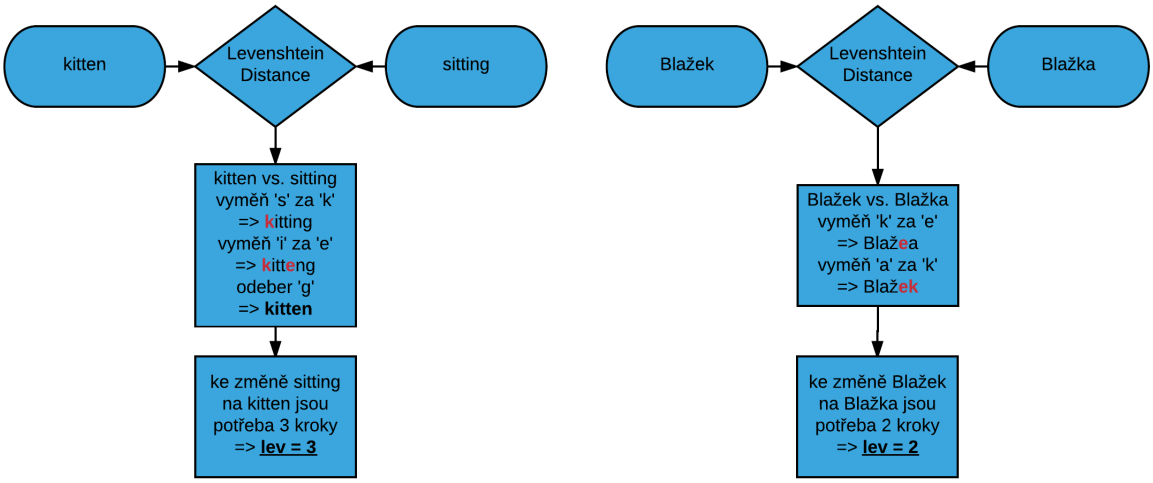
První element v minimu, tedy $lev_{A,B}(i-1,j)+1$, vyplývá z možnosti mazat znak.

Druhý element, tedy $lev_{A,B}(i,j-1)+1$, naopak vyplývá z možnosti přidat znak.

Třetí element, tedy $lev_{A,B}(i-1,j-1)+1_{A_i \neq, B_j}$, vyjadřuje shodu či neshodu znaků na daných pozicích.

$lev_{A,B}(i,j)$ je vzdálenost mezi i-tým znakem textového řetězce `A` a j-tým znakem textového řetězce `B`, kde platí že:

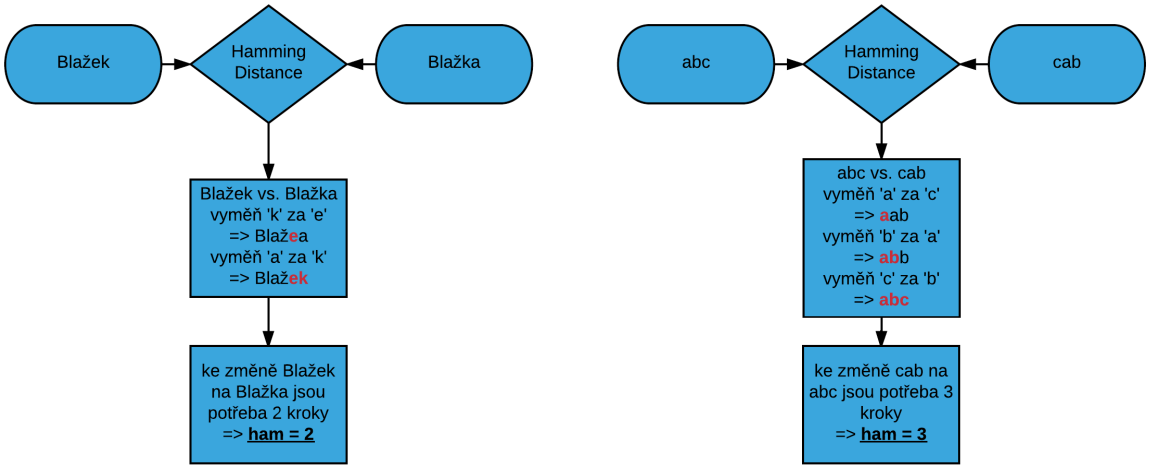
$$1_{A_i \neq, B_j} = \begin{cases} 0, & když \quad A_i = B_j \\ 1, & jindy \end{cases}$$



Obrázek 4.1. Příklad určení Levenshteinovy vzdálenosti

4.2.2 Hammingova vzdálenost

Hammingova vzdálenost, [30] stejně jako Levenshteinova vzdálenost, počítá počet změn v sekvenci znaků `A`, potřebných k získání sekvence znaků `B`. Na rozdíl od Levenshtainovy vzdálenosti je ovšem definovaná pouze za předpokladu, že délka řetězce `A` je stejná jako `B`. Jinými slovy, jako změna se počítá záměna znaku za znak. Vkládání a odstraňování znaků není dovoleno.



Obrázek 4.2. Příklad určení Hammingovy vzdálenosti

4.2.3 Damerau Levenshteinova vzdálenost

Damerau–Levenshteinova [31] vzdálenost také počítá vzdálenost mezi dvěma textovými řetězci. Na základě počtu změn jednoho řetězce, díky kterým vznikne druhý řetězec. V tomto příkladě je kromě změn, dovolených v Levenshteinově vzdálenosti, tedy vkládání, odstranění nebo záměna znaků, dovolena ještě záměna dvou sousedních znaků.

Vztah pro výpočet Damerau–Levenshteinovy vzdálenosti pro dva textové řetězce, vychází ze vztahu pro výpočet Levenshteinovy vzdálenosti (viz. kapitola 4.2.1), který rozšiřuje právě o možnost zaměnit dva znaky.

Opět máme dva textové `A` a `B` s délkami $\|a\|$ a $\|b\|$. Kde $dlev_{A,B}(i,j)$ je vzdálenost mezi i-tým znakem textového řetězce `A` a j-tým znakem textového řetězce `B`.

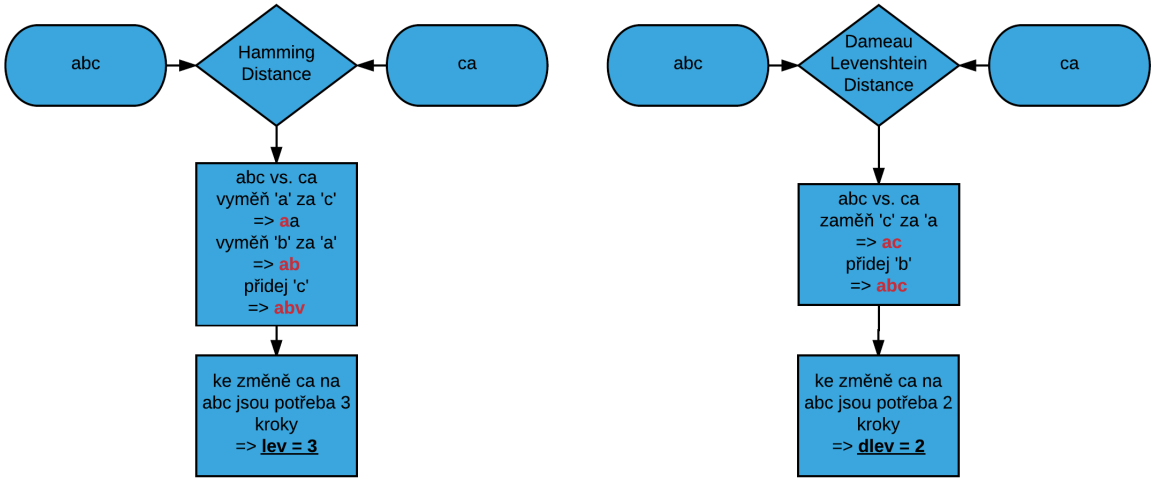
Opět s využitím rekurze můžeme definovat výpočet Damerau–Levenshteinovy vzdálenosti jako:

$$dlev_{A,B}(i,j) = \begin{cases} \max(i,j), & když \min(i,j) = 0; \\ \min \begin{cases} dlev_{A,B}(i-1,j)+1 \\ dlev_{A,B}(i,j-1)+1 \\ dlev_{A,B}(i-1,j-1)+1_{A_i \neq, B_j} \\ dlev_{A,B}(i-2j-2)+1 \end{cases}, & když \quad i > 1 \quad \& \quad j > 1 \\ & \quad \& \quad a_i = b_{j-1} \\ & \quad \& \quad a_{i-1} = b_j \\ \min \begin{cases} dlev_{A,B}(i-1,j)+1 \\ dlev_{A,B}(i,j-1)+1 \\ dlev_{A,B}(i-1,j-1)+1_{A_i \neq, B_j} \end{cases}, & jindy \end{cases}$$

4.2.4 Jaro vzdálenost

Vzdálenost Jaro [17] [32] měří vzájemnou podobnost či rozdílnost. Rozdíl oproti Levenshtainově, vzdálenosti (viz. kapitola 4.2.1) je v tom, že Jaro vzdálenost již naproti vzdálenosti Levenshteinově nevyjadřuje počet potřebných změn v jednom z řetězců, ale jejich podobnost, kde hodnota nula znamená, že vstupní stringy nemají nic společného. Naproti tomu hodnota jedna znamená, že jsou shodné.

Počet rozdílů mezi stringy je vztažen k délce stringů a zároveň není požadavek na to, aby byly znaky textového řetězce `A` přesně na stejných pozicích v řetězci `B`. Stačí, že jsou v okolí určeném parametrem t.



Obrázek 4.3. Příklad porovnání Levenshteinovy vzdálenosti (levá část) a Damerau–Levenshteinovy vzdálenosti (pravá část)

Vztah pro výpočet Jaro vzdálenosti je:

$$dJ_{A,B} = \begin{cases} 0, & když \quad m = 0; \\ \frac{1}{3}\left(\frac{m}{\|s_1\|} + \frac{m}{\|s_2\|} + \frac{m-t}{m}\right), & jindy \end{cases}$$

kde

- m je dáno počtem stejných znaků řetězců `A` a `B` ve stejných oblastech řetězce. Není tedy požadavek, aby řetězce měly přesně ve stejných místech stejné znaky. Stejné znaky musejí být pouze v oblasti jejíž šířka je určena parametrem t, (viz. obrázek 4.4).
- t je dáno jako méně než polovina počtu transpozic. Určíme ji z nerovnice:

$$t = \left(\frac{\max\left(\|s_1\|, \|s_2\|\right)}{2}\right) - 1 \qquad \in N^0$$

4.2.5 Jaro-Winkler vzdálenost

Metoda měření vzdálenosti Jaro-Winkler [17] [32] měří vzdálenost mezi dvěma stringy, neboli jejich vzájemnou podobnost či rozdílnost. Rozdíl oproti Jaro vzdálenosti (viz. kapitola 4.2.4) je v tom, že Jaro-Winkler vzdálenost dává větší důraz na shodu stringů v počátku. Neboli znaky na začátku textového řetězce mají větší váhu než znaky na konci.

Vztah pro výpočet Jaro-Winkler vzdálenosti zapíšeme pomocí výpočtu Jaro vzdálenosti. Vzdálenost Jaro-Winkler je rozšíření výpočtu Jaro vzdálenosti o parametr p, který znakům na začátku řetězce poskytuje vyšší váhu než těm na konci řetězce. Hodnota p se standartně pohybuje v $< 0,1; 0,25 >$.

Jaro-Winckler vzdálenost potom určíme jako:

$$dJW_{A,B} = dJ_{A,B} + lp(1 - dJ_{A,B})$$

kde

- $dJ_{A,B}$ je vypočítaná Jaro vzdálenost
- l je délka společné předpony obou stringů (maximálně však čtyři znaky)
- p konstanta určující váhu počátečních znaků

Můžeme vypozorovat že Jaro-Winkler není ve skutečnosti matematickou vzdáleností, protože nesplňuje trojúhelníkovou nerovnost $d(x,z) \leq d(x,y) + d(y,z)$ a také nesplňuje axiom $d(x,y) = 0 \leftrightarrow x = y$.

Rozdíl mezi hodnotou vzdálenosti Jaro a Jaro-Winkler můžeme pozorovat na obrázku 4.4.

	S	I	T	T	I	N	G
K	0	0	0	0	0	0	0
I	0	1	0	0	0	0	0
T	0	0	1	1	0	0	0
T	0	0	1	1	0	0	0
E	0	0	0	0	0	0	0
N	0	0	0	0	0	1	0

Jaro vzdálenost

t=3 (v tabulce naznačeno světle modrou barvou)

$|s_1| = 7$; $|s_2| = 6$

m = 6

dJ = 1/3*(6/6+6/7+(6-3)/6)

dJ = 0,7857

Jaro-Winkler vzdálenost

p = 0,2 l = 0 (malé L, počet počátečních schodných znaků)

dJW = dJ + lp(1-dJ)

dJW = 0,7857 + 0*0,2*(1-0,7857)

dJW = 0,7857

	B	L	A	Ž	E	K
B	1	0	0	0	0	0
L	0	1	0	0	0	0
A	0	0	1	0	0	0
Ž	0	0	0	1	0	0
K	0	0	0	0	0	1
A	0	0	0	0	0	0

Jaro vzdálenost

t=2 (v tabulce naznačeno světle modrou barvou)

$|s_1| = |s_2| = 6$

m = 5

dJ = 1/3*(5/6+5/6+(5-2)/5)

dJ = 0,7555

Jaro-Winkler vzdálenost

p = 0,2 l = 4 (malé L, počet počátečních schodných znaků)

dJW = dJ + lp(1-dJ)

dJW = 0,7555 + 4*0,2*(1-0,7555)

dJW = 0,9511

Obrázek 4.4. Příklad porovnávající výpočet Jaro vzdálenosti se vzdáleností Jaro-Winkler

4.3 Porovnání editačních vzdáleností

Z uvedeného porovnání editačních vzdáleností (viz. kapitola 4.2) textových řetězců se pro naše účely použití evidentně nejvíce hodí vzdálenost Jaro-Winkler (viz. kapitola 4.2.5). Vzdálenosti, počítající počet rozdílů mezi řetězci jako počet nutných změn znaků, nejsou vhodné, protože stejný počet změn může ukazovat na rozdíl mezi dvěma opravdu rozdílnými slovy, jako například „kitten“ a „sitting“ (viz. obrázek 4.1). Řádově stejný počet změn může být ovšem také zaviněn pouze skloňováním či časováním daného slova jako například „Blažek“ a „Blažka“ (viz. obrázek 4.1).

Problémem je samotný fakt ryze diskrétních hodnot při počítání vzdáleností mezi stringy pomocí například Levenshteinovy vzdálenosti. Tato vlastnost by se dala intuitivně vylepšit například tak, že by byl počet změn podělen délkou textového řetězce. Protože intuitivně cítíme, že je rozdíl mezi tím, zda se textové řetězce liší ve dvou znacích z pěti nebo ve dvou znacích z deseti a podobně.

Vstup	Levenshtein	Damerau–Levenshtein	Hamming	Jaro	Jaro-Winkler
blažek vs. blažka	2	2	2	0,8888	0,9778
michal vs. michala	1	1	-	0,9523	1
zavolat vs. zavolej	2	2	2	0,8095	1
napsat vs. napiš	3	3	-	0,7000	0,8800
hledej vs. vyhledej	2	2	-	0,9166	0,9166
sobota vs. sobotu	1	1	1	0,8888	1
pět vs. pátou	3	3	-	0,6888	0,7511
hodina vs. hodin	1	1	-	0,9444	1
sitting vs. kitten	3	3	-	0,7460	0,7460
abc vs. ca	3	2	-	0,6111	0,6111
sedum vs. osm	4	4	-	0,6888	0,6888
minuta vs. minutník	3	3	-	0,8194	1
zapsat vs. zavolat	3	3	-	0,7460	0,8476
novák vs. nováková	3	3	-	0,8750	1
poznámka vs. procházka	3	3	-	0,7500	0,8000
napsat vs. na	4	4	-	0,7777	0,8666
zapsat vs. za	4	4	-	0,7777	0,8666
sergej vs. s	5	5	-	0,7222	0,7777

Tabulka 4.1. Porovnání metod měření vzdálenosti mezi textovými řetězci

Právě tento typ vylepšení přináší Jaro měření vzdálenosti, které ještě vylepšuje měření Jaro-Winkler, poskytující větší váhu znakům na začátku řetězce.

Z tabulky (viz. tabulka 4.1) můžeme vypozorovat několik dalších vlastností metod měřících editační vzdálenosti. Na první pohled je patrné, jak velkým omezením je skutečnost, že výpočet Hammingovy vzdálenosti (viz. kapitola 4.2.2) je omezen pouze na případ, kdy mají oba textové řetězce stejnou délku.

Dalším zjištěním, vzniklým z pozorování, je skutečnost, že Levenshteinova (viz. kapitola 4.2.1) vzdálenost a Dameau-Levenshteinova (viz. kapitola 4.2.3) vzdálenost vracejí velmi často stejný výsledek, což plyne ze skutečnosti, že rozdílem mezi nimi je pouze schopnost Dameau-Levenshteinovy vzdálenosti zaměňovat sousední znaky. Tato schopnost by svoje uplatnění našla zjevně spíše při opravě překlepů, kde k záměně sousedních znaků dochází relativně často.

Porovnáním hodnot vzdálenosti Jaro a Jaro-Winkler můžeme pozorovat vliv vyšší váhy počátečních znaků u metody měření vzdálenosti Jaro-Winkler. Zavedeme-li funkci α_{JW} nabývající hodnot 0 a 1. Prahovou hodnotu Θ, která určuje hranici mezi skutečností, že si textové řetězce jsou nebo naopak nejsou podobné.

Platí tedy že:

$$\alpha_{JW} = \begin{cases} 1, & dJW \geq \Theta; \\ 0, & dJW < \Theta. \end{cases}$$

Z pozorování vyplývá, že vhodná prahová hranice, pro měření vzdálenosti metodou Jaro-Winkler, je v rozmezí $\Theta =< 0,85; 0,9 >$.

Podobnou hranici bychom se mohli pokusit najít i pro měření vzdálenosti metodou Jaro. Tento práh by byl o něco menší, nabízí se hodnota v rozmezí $\Theta =< 0,8; 0,85 >$. Pohledem do tabulky (viz. tabulka 4.1) ovšem zjistíme, že tímto způsobem bychom nezískali stejně kvalitní funkci α_J.

Výpočet vzdálenosti dvou textových řetězců metodou Jaro-Winkler má i jednu nepříjemnou vlastnost, kterou můžeme pozorovat například na posledních třech řádcích tabulky (viz. tabulka 4.1). Tím, že první znaky mají vyšší váhu se může stát, že i stringy s velmi rozdílnou délkou, budou mít mezi sebou malou Jaro-Winkler vzdálenost. Toto se typicky může stát tak, jak je uvedeno v tabulce 4.1 například pro předložky a bude třeba na to brát zřetel.

Řešením může být buď kombinovat Jaro-Winkler metriku například s Levenshteinovým měřením vzdálenosti nebo ještě jednodušeji hlídat poměr mezi délkou stringů a rozdílem jejich délek.

4.4 Využití editačních vzdáleností

Z předchozí kapitoly vyplývá, že jako hlavní způsob porovnání textových řetězců použijeme metriku Jaro-Winkler. S její pomocí rozšíříme některé standardní metody v Javě pro porovnávání stringů použité, tedy:

- `equal()`
- `contain()`
- `indexOf()`

4.4.1 Metoda equalByJaroWinkler()

Metoda `equal()`[1] se v Javě dá volat od každého objektu a vyjadřuje, ověřuje, skutečnost, že oba objekty jsou si daným způsobem rovny. Bude-li metoda `equal()` zavolána od instance třídy String, bude výsledkem `true` právě tehdy, když si budou oba stringy rovny. Budou-li mít tedy stejný počet znaků a znaky na stejných pozicích budou shodné.

Námi vytvořená metoda `equalByJaroWinkler()` bude mít vlastnosti dříve definované funkce α_{JW} (viz. kapitola 4.3). Metoda `equalByJaroWinkler()` vrátí tedy logickou jedničku právě tehdy, když podobnost mezi stringy $s1$ a $s2$ podle metriky Jaro-Winkler překročí prahovou hodnotu Θ.

```
public boolean equalByJaroWinkler(String s1, String s2,
 double theta){
      JaroWinkler jw = new JaroWinkler();
      double dJW = jw.getSimilarity(s1, s2);

      return dJW>=theta;
}
```

4.4.2 Metoda indexOfByJaroWinkler()

Metoda `indexOfByJaroWinkler()` volaná v Javě od instance třídy string vrací pozici ve stringu $s1$ od které následuje string $s2$. Pokud string $s1$ neobsahuje $s2$, je návratová hodnota rovna -1.

Při definování metody `indexOfByJaroWinkler()` opět využijeme dříve definované funkce α_{JW}. Jestliže je první string $s1$ delší než druhý string $s2$, „prohledáme“ stringem $s2$ za použití metody `equalByJaroWinkler()` string $s1$. Rozdělíme tedy string $s1$ na substringy délky $s2$. Tyto substringy následně otestujeme metodou `equalByJaroWinkler()` a vrátíme pozici, od které se stringy budou shodovat.

[1] `https://docs.oracle.com/javase/7/docs/api/`

Jestliže je string $s1$ kratší než string $s2$, bude výstupem metody `containByJaroWinkler()` v závislosti na výstupu metody `equalByJaroWinkler()` buď 0 nebo -1. Kde hodnota -1 bude indikovat, že $s1$ není podobný $s2$.

```
public int indexOfByJaroWincler(String s1, String s2, double theta) {
    if (s1.length() > s2.length()) {
        JaroWinkler jw = new JaroWinkler();
        for (int i = 0; i + s2.length() <= s1.length(); i++) {
            if (equalByJaroWinkler(s1.substring(i, i + s2.length()),
             s2, theta)) {
                return i;
            }
        }
        return -1;
    } else {
        if (equalByJaroWinkler(s1, s2, theta)) {
            return 0;
        } else {
            return -1;
        }
    }
}
```

4.4.3 Metoda containByJaroWinkler()

Metoda `contain()` volaná v Javě od instance třídy string vrací logickou hodnotu true právě tehdy, když instance, od které je metoda `contain()` volána, obsahuje danou posloupnost znaků. Nemusí být tedy již splněna podmínka stejné délky stringů. Dostačující podmínkou je, že se v textovém řetězci vyskytuje daná posloupnost znaků.

Na tomto místě se nabízí stejný přístup jako v metodě `indexOfByJaroWinkler()`. Tento přístup povede ke správnému výsledku. V rámci minimalizace kódu využijeme metody `indexOfByJaroWinkler()` a metodu `containByJaroWinkler()` plně odvodíme od této metody.

Jestliže `indexOfByJaroWinkler()` vrátí jinou než chybovou hodnotu -1, bude výstupem logická jednička. V případě hodnoty -1 bude výstupem `containByJaroWinkler()` naopak logická nula.

```
public boolean containsByJaroWincler2(String s1, String s2,
 double theta) {
    return indexOfByJaroWincler2(s1, s2, theta)>=0;
}
```

Kapitola 5
Návrh vlastní aplikace

V této kapitole bude představen vlastní návrh hlasového asistenta pro nevidomé uživatele, fungující na mobilním operačním systému Android. Aby uživatelům bylo prozkoumávání aplikace co nejvíce usnadněno, budou její funkce uvolňovány postupně, v několika vlnách. Díky tomu bude získávána zpětná vazba vždy primárně na naposledy uvolněnou část aplikace, a bude se tak lépe reagovat na podněty pro zlepšení atd.

Ve spolupráci se SONC [33] byly identifikovány hlavní funkce, které by měl hlasový asistent pro nevidomé iplementovat, včetně jejich priority. Podle tohoto seznamu bude při práci postupováno.

Protože aplikace bude součástí většího celku, českého launcheru pro nevidomé [34], bude se aplikace zaměřovat primárně na českou, anglickou a německou verzi. Protože většina nevidomých testrů, uživatelů, kteří budou aplikaci moci zkoušet před tím, než se dostane mezi běžné uživatele, jsou Češi, bude aplikace vyvíjena s důrazem na fungování v češtině.

5.1 Fáze vývoje

Samotný vývoj aplikace byl rozdělen do několika nezávislých částí, které byly po jejich dokončení vždy alespoň částečně otestovány (viz. kapitola 6). Pořadí implementace schopností hlasového asistenta vychází z jejich důležitosti, která byla stanovena na základě diskuse se SONS [33] a vývojáři, zabývajícími se podobnými projekty pro nevidomé (viz. kapitola 6.2.1).

5.1.1 Fáze 1

1. Volání - vytáčení kontaktů ze seznamu a čísel, s možností přiřadit si ke kontaktům aliasy (rodinné příslušníky atd. (viz. kapitola 2))
2. Zprávy - posílání texových (SMS) zpráv kontaktům ze seznamu a přímo číslům, opět i s možností aliasů
3. Otevírání aplikací v telefonu
4. Ovládání základních funkcí telefonu (Wi-Fi, Bluethoot, vypnutí telefonu, hlasitost, jas)

5.1.2 Fáze 2

1. Počasí - řekne uživateli jaké bude počasí v dané lokalitě daný den
2. Poloha - řekne uživateli, kde se zrovna nachází
3. Budík - dovolí uživateli si nastavit jednorázový budík na daný čas
4. Odpočet - nastaví a spustí odpočet času

5.1.3 Fáze 3

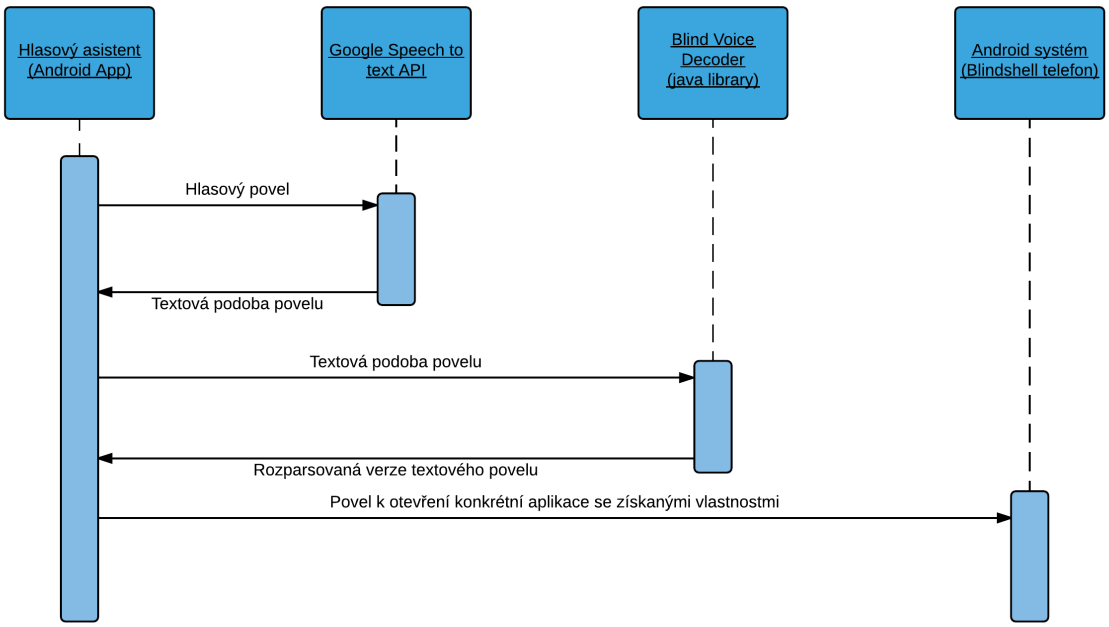
1. Událost - vytvoří v kalendáři událost na dané datum
2. Anekdoty a příběhy - na požádání řekne uživateli vtip nebo krátký příběh
3. Kalkulačka - vrátí výsledek zadaného příkladu

5.1.4 Fáze 4

1. Odpovědi na otázky - vyhledá odpověď na otázku na internetu a vrátí lidskou verzi („ *Jak je vysoká Eiffelova věž?*“, „ *Kdy zemřel Karel IV?*“, „ *Kolik obyvatel má Mexiko?*“ atd.)

5.2 Princip fungování aplikace Blind Voice Decoder

Aplikace bude rozdělena na tři separované části podle následujícího schématu (viz. obrázek 5.1). První částí je Speeech-to-text modul, který zajistí převod mluveného slova do textové podoby. Druhou částí bude JAVA aplikace nazvaná Blind Voice Decoder, která tento textový vstup rozdělí na strojově jednoduše zpracovatelné celky. Třetí částí bude samotná aplikace v telefonu s operačním systémem Android, která bude jednotlivé moduly se získanými parametry na základě výstupu Blind Voice Decoderu spouštět.



Obrázek 5.1. Sekvenční diagram aplikace

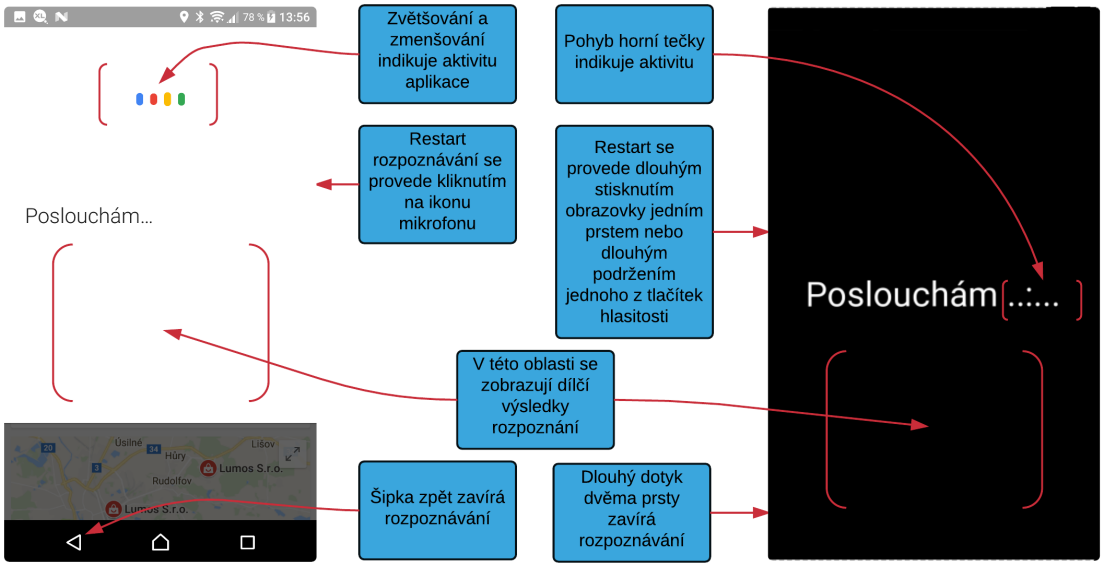
K tomuto rozdělení se dospělo z několika důvodů:

1. Oddělení Speech-to-text modulu plyne ze skutečnosti, že je používáno řešení třetí strany
2. Oddělením části aplikace, sloužící k dekódování textu, je získána možnost vyvíjet tento dekodér samostatně, na základě vstupních souborů, bez nutnosti jej neustále

nahrávat do telefonu. Klíčové bude pouze udržovat jednotný interfece pro komunikaci s Android aplikací.
3. Část aplikce v telefonu bude vypadat jednodušeji, bude pouze spouštět aplikace telefonu a inicializovat decodér. Decodér bude předem otestován na testovacích vstupních souborech.

5.3 Speech-to-text modul a ovládání aplikace

Jak již bylo popsáno v kapitole 3 v aplikaci bude pro převod řeči na text použit Google Cloud Speech API. Způsob použití byl již popsán v kapitole (viz. kapitola 3.3.1), a proto je v tomto místě věnována pozornost výhradně uživatelskému rozhraní. Uživatelské rozhraní vychází z potřeb a možností Google Cloud Speech API a jeho přizpůsobení potřebám nevidomého uživatele bylo konzultováno se střediskem SONS [33].



Obrázek 5.2. Porovnání ovládání Google Speech-to-text a mé aplikace pro nevidomé

K ovládání aplikace se kromě hlasu používají jednoduchá gesta, která jsou nevidomým uživatelům projektu launcheru dobře známa. Jsou to:

- Krátký dotyk jednoho prstu - nevyužit, běžně slouží k procházení menu
- Krátký dotyk dvěma prsty - zopakuje poslední informaci na displeji
- Dlouhý dotyk jedním prstem - potvrzení, v tomto případě slouží ke znovuspuštění hlasového rozpoznávání
- Dlouhý dotyk dvěma prsty - zavírá aplikaci

Aplikaci je možné také otevřít pomocí dlouhého podržení jednoho z tlačítek hlasitosti. Dlouhým podržením se v tomto kontextu rozumí delší než 1s.

5.4 Dekódování získaného textu a jeho parsování

Před samotným začátkem vývoje byl vytvořen trénovací soubor dat pro každý jazyk, na kterém bylo řešení dekódování a parsování testováno. Tato data byla získávána pomocí

dotazníku (viz. příloha C). Takto získaná data jsou součástí přílohy práce (viz. příloha H) a pro potřeby strojového zpracování jsou uložena ve formátu XML v následující podobě

```
<resources>
<sentence id="0">
    <input>Napiš zprávu pro Tomáše Bartáka</input>
    <control>[APP]=Message;[CONTACT]=[Barták Tomáš];
    [TEXT]=null</control>
</sentence>
<sentence id="1">
    <input>Zavolej Tomáše Blažka</input>
    <control>[APP]=Call;[CONTACT]=[Blažek Tomáš]</control>
</sentence>
<sentence id="67">
    <input>Probuď mě v 8:00 odpoledne</input>
    <control>[APP]=Alarm;[TIME]=20:00</control>
</sentence>

...
</resources>
```

Věta sloužící jako vstup je uvozena klíčovým tagem `input`. Tag `control` ukrývá požadovaný správný výstup dekodéru.

Je pochopitelné, že kromě tohoto trénovacího souboru bylo potřeba dodat také soubory, které by představovaly například kontakty získané z telefonu a podobně. Právě z tohoto důvodu vznikly pro každý jazyk další podobné soubory jako `contact.xml`, obsahující kontakty ve velmi podobném formátu, případně „radioStations.xml“, obsahující stanice internetových rádií, nebo `songs.xml`, obsahující potencionální skladby nahrané v telefonu.

Kromě vstupního souboru s trénovacími daty a souborů pomocných potřebuje Blind Voice Decodér ke správné funkčnosti také sadu klíčových slov pro dané aplikace, případně pro uvození některých parametrů, textové vyjádření číslic a podobně. Tento soubor se jmenuje `strings.xml`, jeho název je převzat z konvenčně stejnojmenného souboru v prostředí vývoje Android aplikací, a to právě proto, že tento soubor bude ve finální fázi nahrazen právě souborem `string.xml` v Android aplikaci, která bude tato klíčová slova aplikaci Blind Voice Decoder poskytovat při inicializaci.

Protože bylo předem známo, že postupem času bude aplikace a tedy její textové řetězce překládany do několika dalších jazyků a zároveň bylo známo, že tento překlad bude probíhat pomocí nástroje Pootle[1], jsou klíčová slova uložena v poněkud netypickém formátu.

Klasicky by se nejspíše hodilo pro klíčová slova použít pole, které je běžně pod klíčovým slovem „array“ podporováno v Android vývojovém prostředí a umožňuje pro každý jazyk definovat různě velkou množinu slov, což je právě to, co je v tomto místě potřeba. Byl jsem ovšem upozorněn, že právě Pootle s poly nepracuje vždy korektně a nedovolí překladateli vytvořit více položek, než je v jazyce defaultním. Z tohoto důvodu jsem se rozhodl synonyma klíčových slov ukládat atypicky jako obyčejný string, v kterém budou jednotlivá slova oddělena rovnou čarou „|“. Díky tomuto bude moci překladatel i na Pootle serveru zadat libovolný počet klíčových slov. V aplikaci pak budou podle tohoto znaku rozřezány a uloženy do kolekce. Nadále s nimi bude pracováno stejně jako by byly získány z pole rovnou.

[1] `http://pootle.translatehouse.org`

Ke každému z klíčových slov je navíc možné do hranatých „[0.8]“závorek přidat váhu tohoto slova. Váha určuje, jak moc významná je skutečnost, že se slovo v příkazu našlo a jak moc to ovlivní celkové rozhodnutí. Jestliže není váha vyplněna, je nastavena defaultní hodnota 1.

Příklad takto uložených klíčových slov českém jazyce:

```
<resources>
<string name="app_call_synonyms">volat|volání|volej|zavolat|vytočit|
vytáčej|číslo</string>
<string name="app_message_synonyms">zpráva|sms|zprávu|esemsku|smsku|
textovku|napsat|napiš[0.8]|pošli[0.8]|zprávy|číslo|odepiš|odeslat
</string>
<string name="app_contacts_synonyms">kontakty|kontakt|lidé</string>
<string name="app_other_app_synonyms">další[0.8]|ostatní</string>
<string name="app_setting_synonyms">nastavení</string>
...
</resources>
```

5.4.1 Nalezení cílové aplikace

Dekódování vstupního příkazu je provedeno ve dvou základních krocích. V prvním kroku je hledána aplikace, které se zadaný příkaz týká, a v druhé části je zpřesněn tento výsledek o další informace, které se v příkazu najdou, a to právě v závislosti na tom, jaká aplikace byla nalezena v prvním kroku.

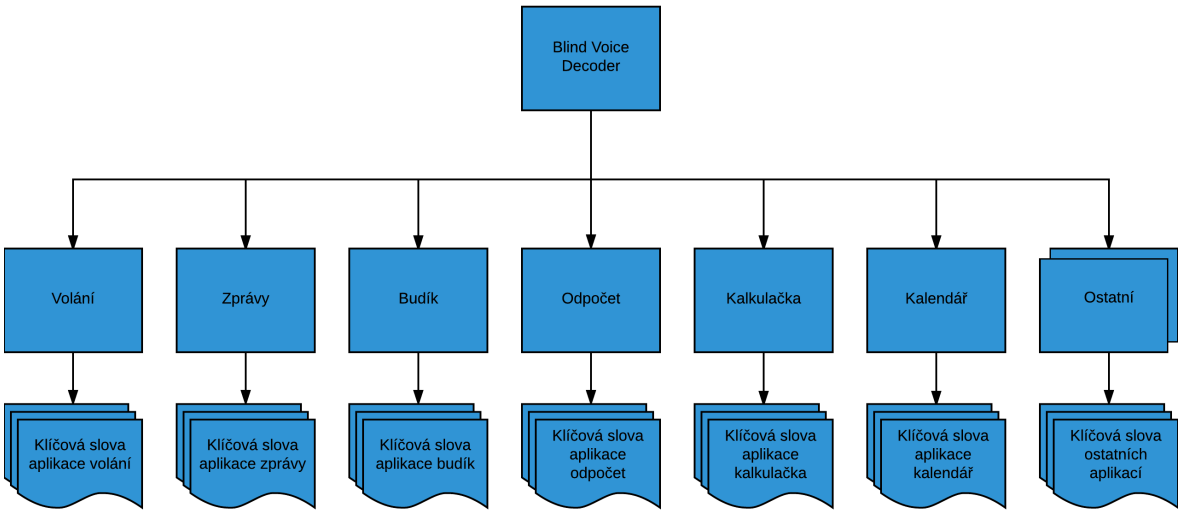
K porovnávání textových řetězců je použit výpočet editační vzdálenosti, popsaný v kapitole 4.2, konkrétně výpočet podle Jaro-Winklera.

Vstupní příkaz je nejprve jednoduše profiltrován několika metodami, které z něj odstraní velmi krátká slova, kratší než dva znaky, a to z toho důvodu, že se jedná o slova, která zpravidla klíčová nejsou. Ovšem alogoritmus Jaro-Winckler dává při výpočtu největší váhu právě na první znaky slova, a tak by se mohlo stát, že tyto (většinou) předložky a spojky by mohly výsledek nepříznivě ovlivnit.

Další použitá metoda filtrování je ta, že slova klíčová pro rozhodnutí o jakou aplikaci má uživatel zájem, se hledají na začátku a konci textového příkazu. Řádově v prvních deseti slovech od začátku i konce (počítáno po odfiltrování velmi krátkých slov). A to prostě proto, že nebyl zjištěn důvod, proč by bylo třeba prohledávat více. Věta, která by měla více než 20 slov, může být v kontextu toho, co decodér umí parsovat pouze snahu napsat zprávu, email, poznámku nebo podobně. A v tomto případě příkaz obsahuje typicky část věty, která slouží právě k označení požadované aplikace a vybrání příjemce a druhou část se samotným textem.

Ačkoli se toto může zdát jako poměrně přísné omezení, během testování se jeho použití opodstatnilo. Díky tomuto opatření totiž například text zprávy nemá vliv na výběr aplikace, což by se jinak stát mohlo. V krajním případě by příkaz vyzývající k napsání textové zprávy, jejíž text by obsahoval výzvu k zavolání někomu jinému, mohl dekodér splést natolik, že by místo vytvoření zprávy mohl rovnou začít vytáčet daný kontakt.

Výsledná aplikace se vybírá na základě skóre získaného porovnáním slov z příkazu a klíčových slov každé aplikace. Každé slovo vstupního příkazu se porovná s každým klíčovým slovem každé aplikace (viz. obrázek 5.3). Toto porovnání se ohodnotí pomocí algoritmu Jaro-Winkler (viz. kapitola 4.2.5) a takto získaným hodnocením (nazvěme jej Ω) se inkrementuje čítač skóre každé aplikace. Protože algoritmus Jaro-Winkler vrací běžně hodnotu okolo $0,5 \in < 0,1 >$, což vzhledem k tomu, že aplikace mají



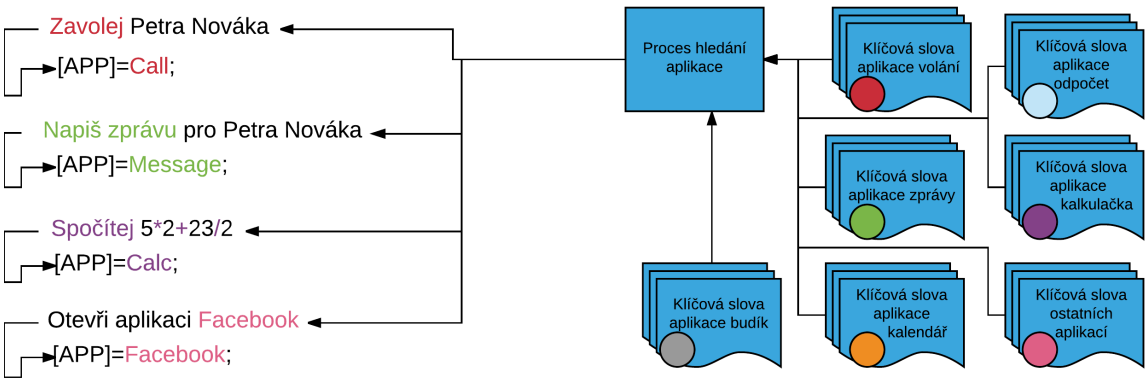
Obrázek 5.3. Principielní diagram aplikace

různý počet klíčových slov může výsledek zásadně ovlivnit. Z tohoto důvodu je v tomto místě zaveden ještě práh ve smyslu, že získaná hodnota musí být větší než práh Θ. Pro inkrement δ poté platí:

$$\delta = \begin{cases} \Omega, & \Omega \geq \Theta; \\ 0, & \Omega < \Theta. \end{cases}$$

Osvědčila se experimentálně nalezená hodnota $\Theta = 0.9$.

Tímto způsobem je nalezena aplikace, která je příkazem žádána (viz. obrázek 5.4). Pokud se v množině vstupních slov nenajde žádná vazba na slova klíčová, pro jednotlivé aplikace tak vyhledávání končí a uživateli se zobrazí oznámení o tom, že mu systém nerozuměl.



Obrázek 5.4. Principielní schéma vyhledávání požadované aplikace (proces rozhodování je barevně naznačen)

5.4.2 Zpřesnění výsledku

Jestliže se podařilo identifikovat požadovanou aplikaci, je čas na zpřesnění výstupu. Informace, která se v textu hledá, je závislá na tom, jaká aplikace byla nalezena v prvním kroku. Je-li identifikovaná aplikace budík, bude se v textu vyhledávat čas, na který se

má budík nastavit. Je-li aplikací kalkulačka, budou se v textu hledat stopy po matematických operacích a číslech, z kterých se posléze poskládá příklad. U aplikací jako jsou volání a zprávy se bude hledat jméno kontaktu, případně text zprávy a tak podobně.

K hledání zpřesňujících informací se využívá opět soubor klíčových slov a Levenshteinův algoritmus Jaro-Winkler k porovnání stringů. Soubor pomocných klíčových slov, tedy těch, které nejsou jmény aplikací, ale slouží k vyhledávání zpřesňujících informací, čítá aktuálně více než 70 konstant, z nichž každá může obsahovat libovolný počet synonym.

Oblasti takto popsané klíčovými slovy, jsou například dny v týdnu, měsíce v roce, číslovky, matematické operace, ale také třeba rodinní příslušníci či slova, sloužící v běžné řeči k uvození věty vložené a podobně. Kompletní seznam je samozřejmě součástí přílohy, zde bude vypsáno jen několik málo příkladů.

```
<resources>
    <string name="app_message_text_start_synonyms">napiš|zprávu|s obsahem
    |kde text je|s textem|ve tvaru|</string>
    <string name="app_note_text_start_synonyms">poznámku|s obsahem
    |kde text je|s textem|ve tvaru|</string>
    ...

    <string name="app_voice_assistant_saturday_synonyms">sobota</string>
    <string name="app_voice_assistant_sunday_synonyms">neděle</string>
    <string name="app_voice_assistant_today_synonyms">dnes|dneska|teď
    |nyní</string>
    <string name="app_voice_assistant_tomorow_synonyms">zítra</string>
    <string name="app_voice_assistant_yesterday_synonyms">včera</string>
    <string name="app_voice_assistant_morning_synonyms">dopoledne
    |ráno</string>
    ...

    <string name="app_voice_assistant_one_synonyms">jedna
    |jednou</string>
    <string name="app_voice_assistant_two_synonyms">dvě|dva
    |druhou</string>
    <string name="app_voice_assistant_three_synonyms">tři|třemi|
    třetí</string>
    ...

    <string name="app_voice_assistant_plus_synonyms">plus|+</string>
    <string name="app_voice_assistant_minus_synonyms">mínus|-</string>
    ...

    <string name="app_voice_assistant_nicknames_mother">mamka|matka|mamča
    |maminka|máma|mámě|mamce|mami|</string>
    <string name="app_voice_assistant_nicknames_father">|otec
    |taťka</string>
    <string name="app_voice_assistant_nicknames_brother">bratr|brácha
    |bráška</string>
    ...
</resources>
```

Účel těchto pomocných klíčových slov je v zásadě dvojí. Za prvé je třeba postihnout co možná nejvíce možností, které by uživatel mohl použít, aby nebyl zbytečně omezován

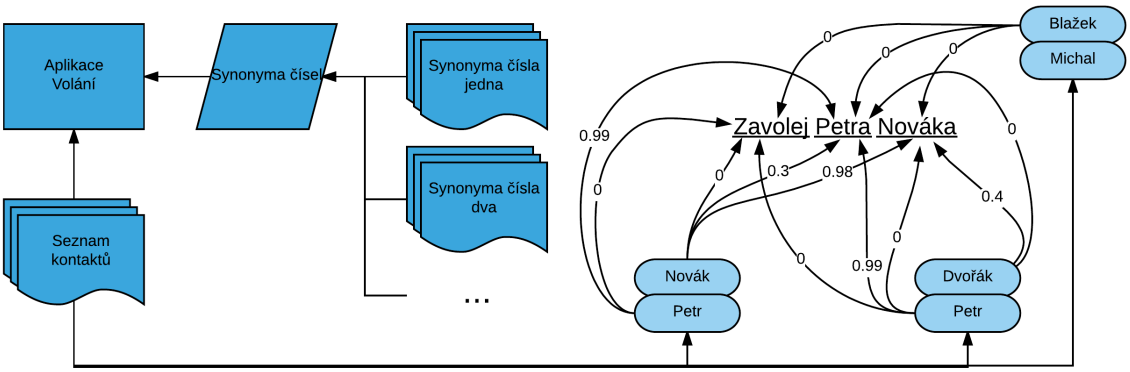
a nemusel mluvit podle předem dané šablony. Druhým úkolem této sady synonym je postihnout co možná nejvíce, ideálně opět všechny tvary, které vrací na výstupu Google Cloud Speech API. V tomto směru není míněno, že by snad Google vracel slova s překlepy nebo nějak obdobně „poškozená“. Toto se týká především diktování číslovek, časových údajů a matematických příkladů, kde se na výstupu Speech to Text API objevují všechny možné formáty, někdy dokonce i ne úplně logicky správné. Jednou se čas napíše ve tvaru „*8:30*“, jindy jako „16 hodin 35 minut“, někdy se po vyslovení například „*osm dvacet dva*“ v textové formě vrátí „ *8:00 22*“ (viz. kapitola 6).

5.4.3 Hledání kontaktů

Prohledat vstupní příkaz a zjistit je-li v něm kontakt a jaký, je potřeba hned v několika případech. Znát kontakt potřebují aplikace, volání, zprávy a email. Princip spočívá v tom, že se každé slovo porovná s křestním jménem i příjmením každého kontaktu. Pro obě jména se opět počítá jakési skóre, které je stejně jako v minulém případě (viz. kapitola 5.4.1) omezeno prahem (viz. obrázek 5.5).

Poté, co se takto ohodnotí všechny kontakty, se znich vybere 10% nejlepších. Zpravidla to znamená, že pokud si kontakty nejsou velmi podobné, nebo pokud uživatel neřekl pouze příjmení nebo pouze jméno, tak se vybere právě jeden kontakt.

V případě, že se najde více než jeden kontakt, ukáže se uživateli seznam s nalezenými kontakty, z kterých si ten správný vybere sám, ručně.



Obrázek 5.5. Principielní schéma zpřesnění výsledku pro aplikaci volání

5.4.4 Hledání čísel

Principiálně opět velmi podobné jako předchozí bod (viz. kapitola 5.4.3). V tomto případě je postup silně závislý na tom, jaký vstupní příkaz přijde z Google Cloud Speech API. Nejjednodušší a nejpřímočařejší možností je telefonní číslo skutečně napsané numerickými číslicemi, typicky po trojicích.

V tomto případě se pouze kontroluje, zdali se jedná o platné telefonní číslo. Kromě čísel záchranných složek, která jsou typicky tříčlenná, se berou všechna čísla s více než čtyřmi číslicemi a méně než čtrnácti (pro případ předvolby 00420 atp.), případně méně než třinácti, začíná-li číslo symbolem plus.

Google Cloud Speech API může ovšem číslo vrátit také napsané slovy či kombinací čísel a slov. Z tohoto důvodu je pro hledání číslic k dispozici slovník, obsahující číslice psané slovem, díky čemuž je vstupní příkaz prohledán a opět pomocí metody porovnávání stringů Jaro-Winkler a prahování je zjišťováno, zdali věta obsahuje psaná

čísla. Takto nalezené číslo je opět zkontrolováno, aby se potvrdilo, že se jedná o platné telefonní číslo.

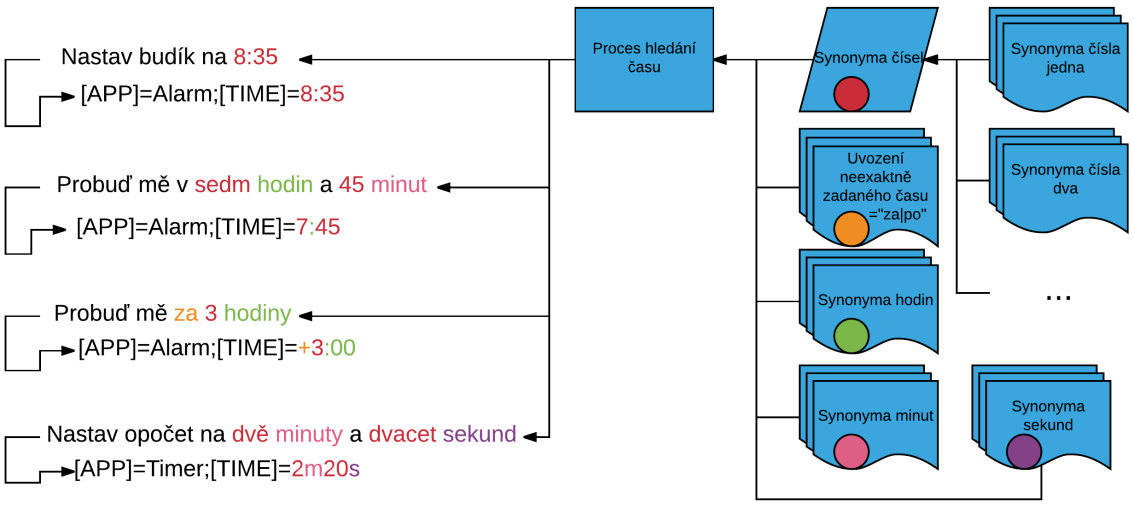
5.4.5 Hledání času

Pro potřeby aplikací, které jsou založeny na měření času, jako jsou budík, stopky či odpočet času, ale také například kalendář, je třeba získat z příkazu časový údaj, tedy pochopitelně pokud jej příkaz obsahuje.

V tomto směru stojí za rozlišení dvě situace. První přímočará situace nastává ve chvíli, kdy je příkazem čas určen jednoznačně, přesně, bez vztahu k současnosti, např.: „*v 8:35*“ a podobně. Druhou situací je určení na základě času aktuálního, tedy typicky např.: „*za tři hodiny*“.

S ohledem na tyto dvě potencionálně možné situace je vyhledávání času řešeno následovně. Nejprve se ve větě hledá samotný časový údaj. Obdobně jako bylo popsáno v kapitole 5.4.4 se vyhledávají čísla psaná jak numericky, tak slovem. Vzhledem k okolnostem se ještě hledají klíčová slova, prozrazující, zda se jedná například o hodiny, minuty, sekundy a podobně. Princip hledání je zachycen na přiloženém obrázku (viz. obrázek 5.6).

Po nalezení samotného časového údaje se řeší otázka, zdali se jedná o první nebo druhou výše popsanou situaci, a to hledáním slova uvozujícího situaci druhou. Tedy tu, že čas není zadán exaktně, ale vztahuje se k současnosti.



Obrázek 5.6. Principiální schéma vyhledávání časového údaje v příkazu (proces rozhodování je barevně naznačen)

5.4.6 Hledání datumu

Nalezení data ve větě se využívá pro potřeby aplikací jako jsou předpověď počasí a kalendář. Nejjednodušší možný scénář je ten, že uživatel v příkazu datum skutečně přímo uvede. Příkaz potom může vypadat například následovně:

Jaké bude počasí 18.4.?

Získat z tohoto příkazu správně datum je obdobné jako získat z příkazu časový údaj (viz. kapitola 5.4.5). Tento způsob dotazu na předpověď počasí je ovšem běžný a přirozený hlavně pro data, která jsou více než týden vzdálena.

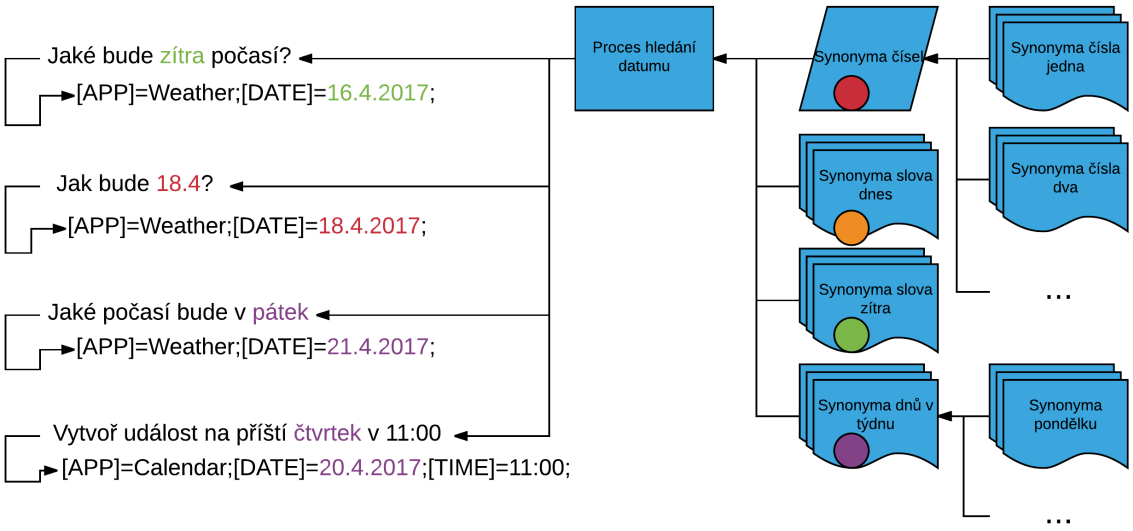
Je-li ovšem hypoteticky 18.4. právě zítra, bude patrně mnohem častější příkaz ve stylu:

Jaké bude zítra počasí?

Obdobným způsobem se typicky ptáme v horizontu řádově jednoho, či dvou týdnů. Z tohoto důvodu se pro nalezení správného data ve větě používá sada synonym, které obsahují jednotlivé dny v týdnu a podobně. Budeme-li mít příkaz:

Jaké bude počasí v pátek?

Blind Voice Decodér nejprve zjistí že aplikace, pro kterou je příkaz určen, je počasí, a následně v příkazu začne hledat datum. Datum napsané formátem DD.MM.YYYY a podobně zjevně neobjeví, ale slovo „*pátek*“ ano. Poté již stačí vrátit datum následujícího pátku.



Obrázek 5.7. Principiální schéma vyhledávání data v příkazu, s předpokladem že dnes je 15.4.2017 (proces rozhodování je barevně naznačen)

V této části je také třeba postihnout rozdílnosti v psaní datumů v jednotlivých jazycích. Pro český jazyk je to formát DD.MM.YYYY, který se užívá například i v němčině a britské angličtině. Naproti tomu v americké angličtině je datum standardně psáno ve formátu MM.DD.YYYY, na což je třeba reagovat.

5.4.7 Hledání matematických operací a příkladů

Vyhledávání matematických příkladů v textu opět vychází z principu vyhledávání čísel v textu (viz. kapitola 5.4.4). Kromě čísel se ovšem ještě hledají matematické operátory, v současné verzi pouze ty základní, tedy plus, mínus, krát a děleno.

Následně se kontroluje logická správnost nalezeného příkladu, tedy například to, že operátory krát a děleno jsou vždy mezi dvěma čísly. I v tomto případě bylo nalezeno několik težkostí, způsobených různými variacemi výstupu Google Cloud Speech API. Například to, že jako operátor násobení může sloužit jak znak „*“ tak i „x“, nebo fakt, že v některých případech je příkaz ve tvaru „číslo krát číslo“ nahrazen posloupností násobné číslovky a čísla. Například „*3 krát 254*“ vrátí Google jako „*třikrát 254*“. Naproti tomu „*3 krát 5*“ vrátí jako „*3 krát 5*“. Tyto skutečnosti byli naštěstí odhaleny a vyřešeny během testování (viz. kapitola 6).

5.4.8 Hledání těla zprávy, textu poznámky a podobně

Hledání textu, který uživatel chce vložit do těla zprávy, se ukázalo jako nejtěžší část implementace, což dopředu indikovala i skutečnost, že si s touto disciplínou ne vždy zcela správně poradí i hlasoví asistenti, za kterými stojí firmy jako Apple, Google či Microsoft (viz. kapitola 2).

Vzhledem k tomu, že není žádoucí, aby byl uživatel omezován a nucen diktovat příkaz pro psaní zprávy, poznámky nebo podobně nějakou předem danou formou, je důležité objevit co nejvíce, ideálně opět všechny, možné scénáře, kterými se dá tento požadavek lidsky zadat. Takto získané scénáře je následně nutné natolik zobecnit, aby postihovaly co možná největší skupinu hlasových příkazů, které budou uživatelé používat. K ověření tohoto samozřejmě poslouží zpětná vazba během testování aplikace (viz. kapitola 6).

Získat tělo textové zprávy nebo emailu je složitější, než získat text poznámky, nebo text k vyhledání na internetu a podobně. Proto se kapitola nadále bude věnovat právě tomuto problému. Získání textu pro poznámku nebo vyhledávání na internetu je zjednodušenou verzí následujícího řešení.

Z příkazu k napsání zprávy potřebujeme získat postupně tři informace. Zaprvé je třeba zjistit, o jakou aplikaci se jedná (viz. kapitola 5.4.1), za druhé najít kontakt (viz. kapitola 5.4.3) nebo telefonní číslo (viz. kapitola 5.4.4) a následně text, pokud jej tedy příkaz vůbec obsahuje.

Zjednodušíme-li celou situaci, můžeme prohlásit, že to, co nám po hledání aplikace a kontaktu nebo telefonního čísla zbyde, je právě text zprávy. Problémem tohoto tvrzení je, že kontakt i text zprávy může být uvozen něčím, co do těla zprávy nepatří. Což je problém, kterým trpí Google Now i Siri (viz. kapitola 2).

V tomto místě je využita drobná újma na obecnosti, která ovšem vychází z vlastností českého, anglického i německého jazyka. Není, zdá se, možné sestavit větu, která by zněla přirozeně a obsahovala tři výše popsané části, jiným způsobem než tím, že část označující použitou aplikaci bude z těchto tří částí na prvním místě. Části odkazující na kontakt či tělo zprávy, jsou zaměnitelné ovšem pouze na druhé a třetí pozici. I tento předpoklad se může zdát omezující, ale během testování se ho nepodařilo vyvrátit.

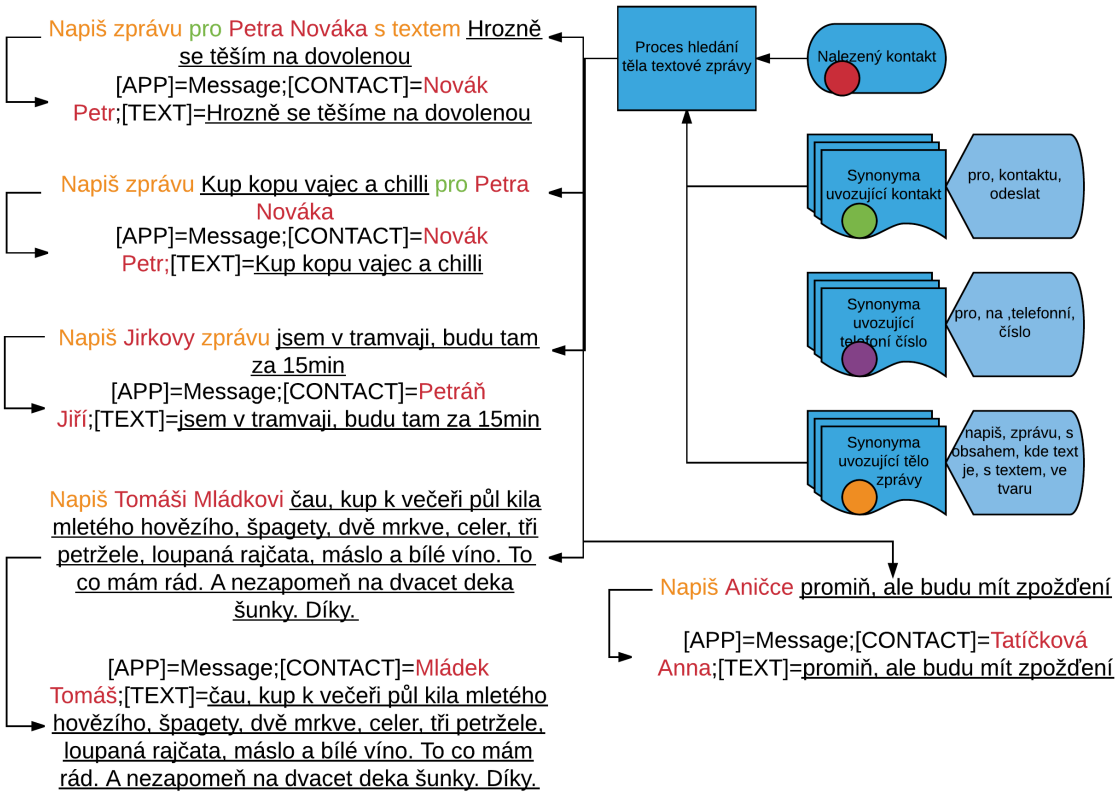
Kontakt byl již nalezen v předchozím kroku (viz. kapitola 5.4.3), teď jen zbývá zjistit jeho polohu ve větě, případně zdali, není uvozen serií některých klíčových slov k tomu sloužících. V případě, že ano, nastavit začátek části věty, obsahující kontakt, s ohledem na toto uvození.

Obdobně se na základě synonym klíčových slov, používaných k uvození zprávy, pokusíme zjistit, zdali ve zprávě jsou, případně na jakém místě. Protože k porovnávání stringů je používán prahovaný algoritmus Jaro-Winkler, není třeba mít mezi synonymy vyjmenované všechny tvary, pády a časy všech slov, připadajících v úvahu.

Díky tomuto přístupu byl vstupní příkaz bezezbytku rozdělen na tři části. Protože je zřejmé, že na začátku příkazu je část věty, která určuje o jakou aplikaci se bude jednat. Následně, v libovolném pořadí, části věty, které odkazující na kontakt či text. Teď již skutečně stačí vybrat z těchto dvou částí tu, která na kontakt nebo číslo, neodkazuje a odříznout z ní uvozující část.

5.5 UML schéma aplikace Blind Voice Decoder

UML schéma aplikace Blind Voice Decoder je znázorněno na přiloženém obrázku (viz. kapitola 5.9). Ve schématu je k vidění i třída Test.java, která slouží, jak její název napovídá, k testování aktuální implementace. Výstupem aplikace je porovnání předem



Obrázek 5.8. Principielní schéma vyhledávání těla zprávy ve větě (proces rozhodování je barevně naznačen)

vytvořeného správného výstupu s výstupem aktuálně nalezeným třídou DecodeCommand.java.

Pakliže se ve větě objeví cílová aplikace (viz. kapitola 5.4.1), a jestliže je u této aplikace nutné hledat další dodatečné informace (viz. kapitola 5.4.2), je od třídy DecodeCommand.java vytvořen potomek, který řeší právě toto zpřesnění. Důvodem této dědičnosti je především rozdělení vlastního kódu do několika souborů pro lepší udržovatelnost a rozšiřitelnost.

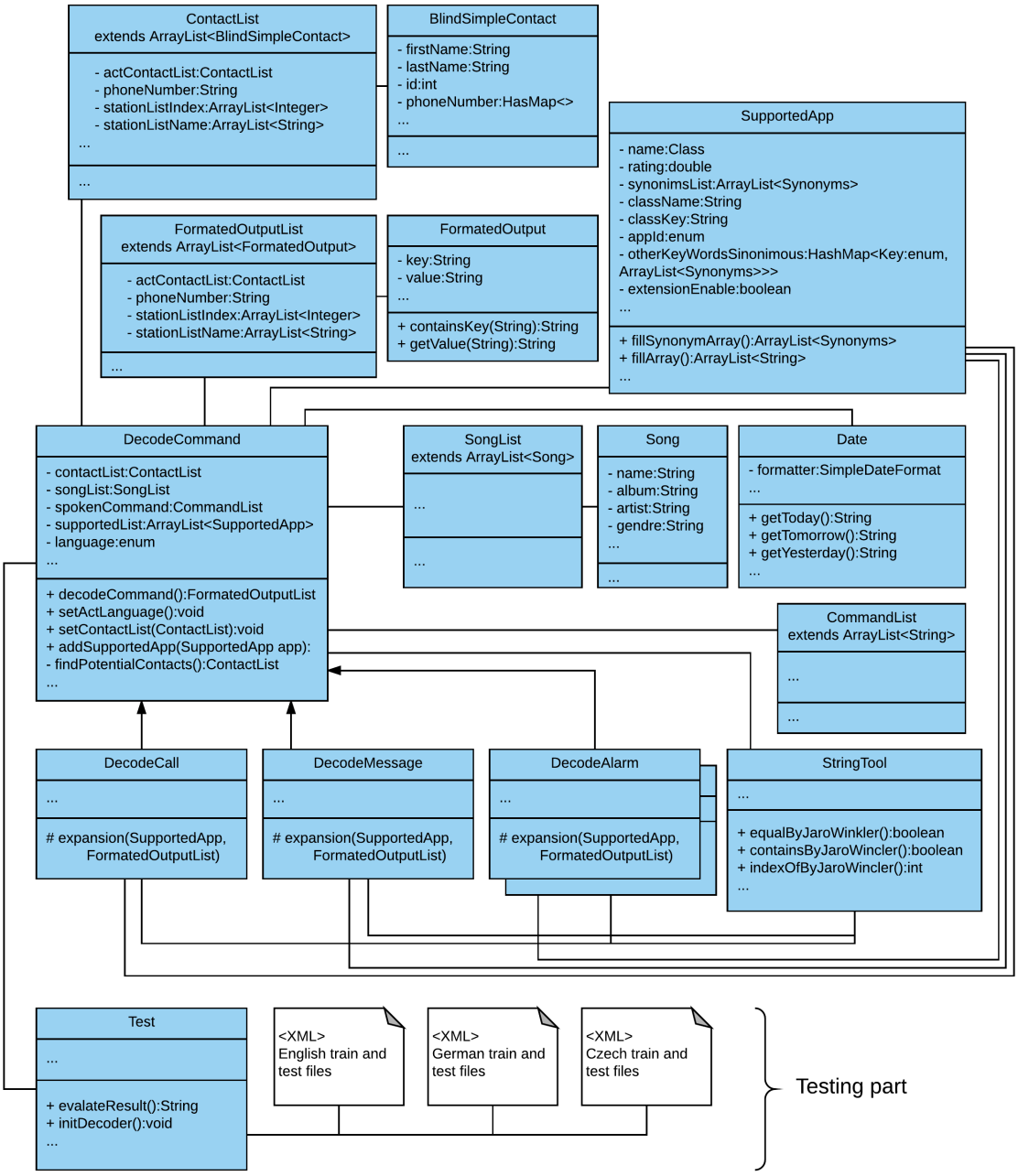
Třída StringTool sdružuje metody pro práci s textovými řetězci jako například metody `equalByJaroWinkler()`, `indexOfByJaroWinkler()` a `containByJaroWinkler()` (viz. kapitola 4.4).

5.6 Vlastní android aplikace

Jak je patrné ze sekvenčního diagramu (viz. kapitola 5.1), samotná android aplikace zastává funkci jakéhosi cetrálního uzlu v celém procesu, začínající hlasovým příkazem uživatele a končící vykonáním nalezené akce na mobilním telefonu.

Implementací Google Speech to Text API se zabývala kapitola 3.3.1 a jejím ovládáním pro nevidomé a slabozraké kapitola 5.3, proto již v tomto místě nebude znovu diskutována.

Implementace knihovny Blind Voice Decoder v prostředí vlastní android aplikace vyplývá z jejího UML schématu (viz. kapitola 5.5) a sekvenčího diagramu (viz. kapitola 5.1). V UML diagramu můžeme pozorovat, že z pohledu volající android aplikace je



Obrázek 5.9. UML schéma aplikace Blind Voice Decoder

třeba Blind Voice Decoder pouze inicializovat a následně mu metodou `ecode(String command)`, třídy DecodeCommand, předat větu k analýze. Inicializace v tomto případě zahrnuje výběr jazyka, v kterém bude následně věta předána a vytvoření instancí podporovaných aplikací včetně jejich klíčových slov.

Hlavní chvíle android aplikace tedy přichází v momentě, kdy obdrží strojově čitelný výsledek z Blind Voice Decodéru. Tvar výstupu Blind Voice Decodéru je v duchu tabulky (viz. tabulka 5.1).

Jak je patrné z UML schématu (viz. kapitola 5.5), k vytvoření takto strukturovaného výstupu slouží třída FormatOutput, a proto tato třída také obsahuje metody pro zjištění

Vstup	Výstup
Zavolej Tomáše Blažka	[APP]=Call;[Contact]=Tomáš Blažek;
Vytoč Tomáše	[APP]=Call;[CONTACT]=[Blažek Tomáš, BartákTomáš, Novák Tomáš, Mládek Tomáš];
Napiš zprávu pro Tomáše Blažka s textem Hrozně se těším na dovolenou	[APP]=Message;[CONTACT]=[Blažek Tomáš]; [TEXT]=Hrozně se těším na dovolenou
Jak bude zítra	[APP]=Weather;[DATE]=2017-05-03
Nastav budík na 8:00	[APP]=Alarm;[TIME]=8:00;
Probuď mě za 3 hodiny	[APP]=Alarm;[TIME]=00:14;
Nastav odpočet na tři minuty	[APP]=Timer;[TIME]=3m;
Spočítej 3 plus 2 krát 4 plus 1 děleno 3	[APP]=Calc;[EXAMPLE]=3+2*4+1/3;
Vytvoř událost Rande s Danem v Raclette na příští úterý ve čtyři odpoledne	[APP]=Calendar;[TEXT]=Rande s Danem v Raclette;[DATE]=2017-05-09;[TIME]=16:00;
Otevři lupu	[APP]=MagnifyingGlass;
Najdi Jak hrál včera manchester united	[APP]=InternetBrowser;[TEXT]=jak hrál včera manchester united;

Tabulka 5.1. Příklady vstupů a výstupů aplikace Blind Voice Decoder

toho, jaké klíčové parametry (flagy) takto parsovaný textový výstup obsahuje. Součástí jsou také metody sloužící k jednoduchému získání hodnot těchto flagů.

Samotné vyvolání cílových android aplikací je pak, v souladu s důrazným doporučením Googlu v dokumentaci Androidu [14], vždy pomocí předání klíčových hodnot pomocí kontejneru Bundle pod definovanými klíči. Tento kontejner si potom aktivita vlastní aplikace vyzvedne a získá z něj potřebné informace pro své fungování, jako například kontakt, text zprávy nebo třeba čas, na který má nastavit budík. Příklad spuštění aktivity pro psaní SMS zprávy:

```
private void startSMSToNumberActivity(String phoneNumber,
        String text) {
    Intent mIntent = new Intent(getApplicationContext(),
        BlindMessagesWriteSMSToContactActivity.class);
    mIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    mIntent.putExtra(VOICE_ASSISTANT_IDENTIFICATION_KEY,
        VOICE_ASSISTANT_IDENTIFICATION_CODE);
    mIntent.putExtra(VOICE_ASSISTANT_LAST_PARSE_OUTPUT_KEY,
        actFormatedOutputList.toString());
    mIntent.putExtra(VOICE_ASSISTANT_PHONE_NUMBER_KEY, phoneNumber);
    mIntent.putExtra(VOICE_ASSISTANT_SMS_TEXT_KEY, text);
    mIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
        | Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(mIntent);
}
```

Protože primární uživatelé budou nevidomí, je každý krok ozvučen. A cesta ke skutečnému vykonání hlasem zadaného pokynu je zpravidla zakončena vyžádaným manuálním potvrzením uživatele. Tato aktivita je vyvinuta z důvodu eliminace nežádoucích akcí, vzniklých z nesprávného pochopení příkazu.

Uživatel je tedy typicky vyzván například k výběru požadovaného kontaktu z několika nejlépe ohodnocených kandidátů. Tento stav typicky nastane hlavně tehdy je-li v telefonu více kontaktů s podobnými jmény, křestními i příjmeními, nebo ve chvíli,

kdy uživatel v příkazu uvede jméno pouze jedno, tedy například pouze křestní. Stejně tak je uživatel vyzván k potvrzení správně nalezeného času budíku či data u události v kalendáři atp.

Obrázek 5.10. Případ, kdy není jisté, který kontakt je požadován

Obrázek 5.11. Potvrzení vytváření budíku

Obrázek 5.12. Vytváření události v kalendáři

Obrázek 5.13. Spouštění konkrétní rádiové stanice

5.6.1 Aktualizace aplikace

Existují dvě možnosti jak vylepšit výstup aplikace po zjištění problému. V případě objevení závažné chyby, tedy buď takové, která způsobuje zhroucení samotné aplikace nebo takové chyby, která ukazuje, že je třeba změnit vyhledávací mechanizmus, je třeba aplikaci aktualizovat klasickým způsobem. V tomto případě tedy nainstalovat nové apk s novou, opravenou verzí aplikace.

Chyby ovšem mohou být způsobeny také „pouze“ chybějícím synonymem mezi klíčovými slovy či jeho chybně nastavenou vahou (viz. kapitola 5.2). V tomto případě se zdá zbytené obtěžovat uživatele s přeinstalováním aplikace, a proto je aplikace vybavena schopností si pravidelně kontrolovat a stahovat aktuální synonyma klíčových slov z webového servru. Vzhledem k návrhu aplikace tento způsob umožňuje značnou úpravu chování celého vyhledávacího mechanizmu, a to bez jakékoli interakce s uživatelem. Tímto způsobem je například možné i absolutně deaktivovat některou z vyhledávaných aplikací.

Kapitola 6
Testování

Aplikace byla testována třemi rozdílnými skupinami uživatelů, a to v různých fázích vývoje. První skupinou uživatelů byli alfatestři, tedy lidé vyvíjející aplikace pro nevidomé a pohybující se v tomto prostředí, často s několikaletými zkušenostmi v oboru.

Druhou skupinou uživatelů jsou technicky velmi zdatní, ovšem již nevidomí lidé, kteří mají dlouholeté zkušenosti právě s testováním pomůcek pro nevidomé a slabozraké a jsou schopni se lépe vcítit do potřeb budoucích potencionálních uživatelů. Také dokáží oddělit svá osobní privilegia a přednosti a odhadnout, co bude vyhovovat větší části budoucích nevidomých uživatelů.

Třetí, poslední, skupinou uživatelů, kteří byli zapojeni do testovní, byli přímo budoucí nevidomí uživatelé, kteří byli s ovládáním a možnostmi implementace v rychlosti seznámeni a následně aplikaci testovali.

Testování aplikace bylo poměrně dlouhodobé, trvalo řádově 9 měsíců, podle toho, jak byly postupně dokončovány jednotlivé fáze vývoje (viz. kapitola 5.1). Díky dlouhodobému testování mohlo být několikrát reagováno na náměty, vzniklé z testování.

6.1 Zpětná vazba

Zpětná vazba byla získávána různými způsoby od jednotlivých skupin testujících uživatelů. Zpětná vázba byla získávána jak při osobních setkání, tak telefonickou nebo elektronickou komunikací s testujícími uživateli. Druhá a třetí skupina uživatelů navíc před a po testování vyplnila krátké dotazníky, které jsou součástí přílohy (viz kapitoly D a E) a jejichž výsledek bude diskutován v této kapitole.

Jak již bylo uvedeno, aplikace se skládá ze tří částí (viz. kapitola 5.1). A proto byla zavedena důkladnější zpětná vazba, než jen komunikace s jednotlivými skupinami uživatelů. Androidí část aplikace obsahuje možnost ukládat získané údaje z používání aplikace do MySQL databáze. Data byla naprosto anonymizována a byla sbírána s plným vědomím testujících uživatelů. Díky této zpětné vazbě byla nasbírána další data, která byla přidána do testovacího setu, sloužícího k ověřování funkčnosti a schopností aplikace Blind Voice Decoder (viz. kapitola 5.2).

Tato zpětná vazba obsahovala datum a čas, kdy byl záznam uložen, aby se dalo odhadnout, zda uživatel například opravdu požívá poslední verzi aplikace a není tedy možné, že potencionální problém je již vyřešen. Ukládán byl také výstup Google Cloud to Speech API, kterým byla rozšiřována testovací sada aplikace Blind Voice Decoder (viz. kapitola 5.2). Následně byl ukládán výstup aplikace Blind Voice Decoder, tedy nalezená aplikace a případně zpřesňující informace, získané z uživatelova příkazu.

Také díky této zpětné vazbě byly zjistěny zvláštnosti a nepřesnoti, které se občas vyskytují ve výstupu Google Cloud Speech to Text API (viz. kapitola 3.3.1) a které budou diskutovány dále.

6.2 Testovací fáze

Testování aplikace bylo rozděleno na obvyklé tři fáze. Do první fáze, alfa testování, byli zapojeni jen a pouze vývojáři, podílející se na podobných projektech pro nevidomé a zrakově postižené. V druhé fázi, beta testování, se k alfa testrům připojili technicky velmi zdatní uživatelé z řad nevidomých, kteří byli schopni poskytnout neobyčejně kvalitní zpětnou vazbu. Třetí fáze probíhala s pomocí uživatelů, kteří do vývoje nebyli nijak zapojeni, a to jak z řad nevidomých a slabozrakých, tak z lidí z autorova okolí.

6.2.1 Alfa testování

První fáze testování probíhala převážně za účasti lidí, kteří sami mají zkušenosti s vývojem aplikací pro operační systém Android pro nevidomé a slabozraké. Cílem a úkolem tohoto testování bylo primárně ověřit funkčnost ovládání android aplikace a odladit stabilitu. Zejména pak schopnost restartovat Google Speech to Text API. Během testování si aplikace a hlavně telefony sami prošly skutečným peklem a i díky tomu se podařilo objevit spoustu nedostatků, způsobujících často „zamrznutí“ či zhroucení aplikace.

Toto testování probíhalo již ve chvíli, kdy byla implmementována první fáze vývoje aplikace (viz. kapitola 5.1.1). Nebyly tedy implementovány všechny zamýšlené schopnosti, což ovšem nebylo překážkou právě proto, že cílem bylo ověřit stabilitu androidí části aplikace. Během této části se testující uživatelé zaměřovali hlavně na český jazyk.

6.2.2 Beta testování

Cílem beta testování bylo již primárně ověřovat schopnosti rozpoznání hlasových povelů uživatelů a po jednotlivých částech kontrolovat vývojové fáze (viz. kapitola 5.1). V této fázi neocenitelnou roli sehrálo ukládání rozpoznaného textu a výstupu aplikace Blind Voice Decoder do MySQL databáze (viz. kapitola 6.1). Tato skutečnost velmi usnadňovala práci, protože bylo možné sledovat, jakým způsobem probíhá interakce mezi uživateli a telefonem. Typická stížnost uživatele totiž byla ve tvaru:

Nepovedlo se mi nastavit budík na 8:35.

A právě díky datům ukládaným do databáze bylo možné chyby relativně rychle opravovat a řešit. Z uložených informací velmi rychle vyplynulo v jaké ze tří částí hlasového ovládání (viz. kapitola 5.2) k chybě došlo. V této části testování bylo objeveno několik nepředpokládaných výstupů Google Cloud Speech API. Několik příkladů je uvedeno v přiložené tabulce (viz. tabulka 6.1).

Uživatelův příkaz	Výstup Google Coud Speech to Text API
„Nastav budík na 8:30“	nastav budík na 8:30
„Nastav budík na 8:32“	nastav budík na 8:30 2
„Spočítej 5+7“	spočítej 5 + 7
„Spočítej 5/7“	spočítej 5 děleno 7
„Spočítej 5*7“	spočítej 5 x 7
„Spočítej 3*7“	spočítej třikrát 7
„Spočítej 300*7“	spočítej 300 * 7
„Spočítej 3000000*7“	spočítej 3 milóny krát 7

Tabulka 6.1. Příklady nedokonalosti výstupu Google Coud Speech to Text API

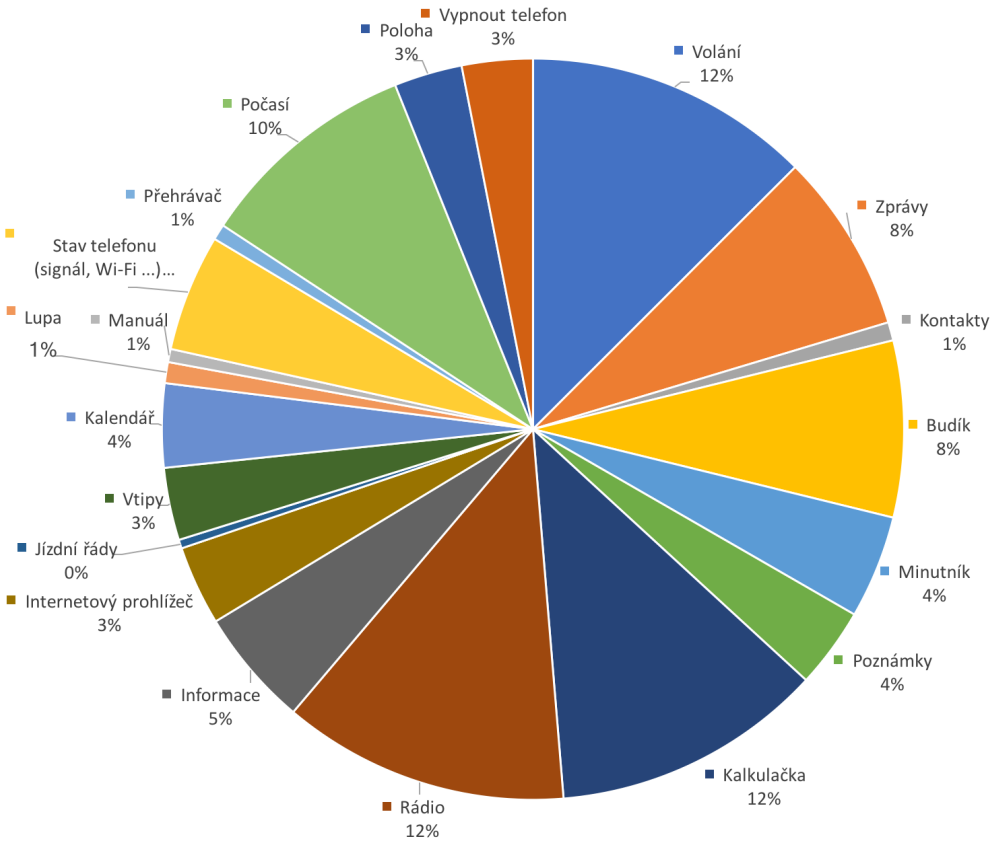
Tyto neduhy byly vyřešeny dodatečnými filtry, které v rámci možností tyto nepřesnoti eliminovaly a výstup tím normalizovaly. Tyto filtry byly využity během vyhledávání zpřesňujících informací v aplikaci Blind Voice Decoder.

Beta testování se již netýkalo pouze českého jazyka, ale také angličtiny a němčiny. Díky ukládání zpětné vazby a tomu, že víme, kdo přesně a kdy měl k aplikaci přístup, můžeme určit kolik uživatelů a řádově jak dlouho aplikaci v daném jazyce testovalo, což dokumentuje následující tabulka (viz. tabulka 6.2).

Jazyk	Počet uživatelů	Počet dotazů	Čas testování
Čeština	13	1131	18,1 h
Angličtina	11	1965	32,4 h
Němčina	4	280	1,2 h

Tabulka 6.2. Statistika beta testování

Testování se tedy v této fázi zúčastnilo 21 uživatelů (někteří testovali ve více jazycích), kteří v uvedených třech jazycích v období od 1.2. do 1.5.2017 položili hlasovému asistentovi celkem 3376 dotazů či příkazů. Bohužel není ze získaných dat možné přesně zjistit úspěšnost rozpoznávání, protože to, jestli se požadovaná akce skutečně vykonala tak, jak uživatel před vyslovením svého příkazu doufal, může posoudit právě jen a pouze tento uživatel a takovýto druh zpětné vazby nebyl při testování zaveden. Částečnou představu o úspěšnosti získáváme až z dat z potestovacího dotazníku (viz. kapitola 6.2.5).



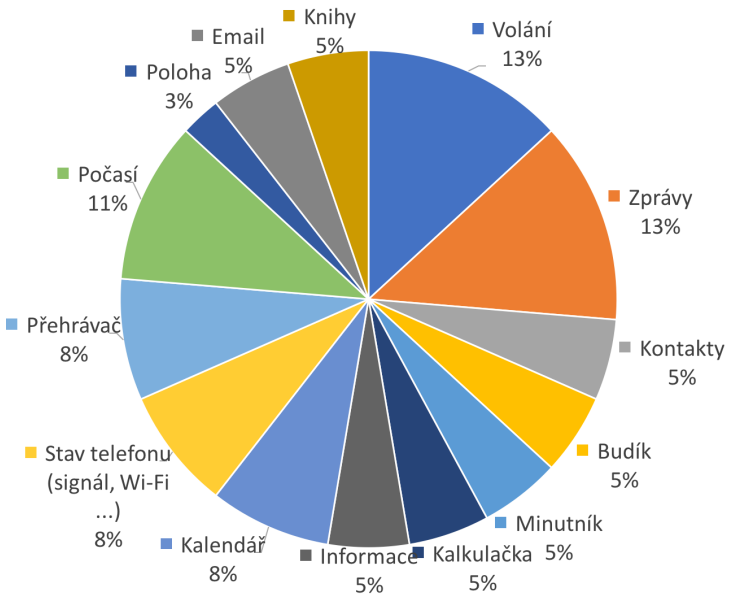
Obrázek 6.1. Statistika vyhledávaných aplikací

6.2.3 Testování na běžných uživatelích

Testování na nevidomých uživatelích, kteří nebyli přímo zapojeni do vývoje, probíhalo ve spolupráci se SONS [33]. Informace z této fáze testování byly získávány hlavně pomocí předtestového (viz. příloha D) a potestového dotazníku (viz. příloha E). Dotazníky vyplnilo pouze 6 lidí, což obecně není statisticky významný vzorek, protože se ovšem odpovědi na položené otázky zásadně nerozcházejí, je možné i s tímto množstvím získaných informací nadále pracovat. Cílem této fáze testování bylo ověřit funkčnost implementace a případně zjistit o jaké, dopusud neimplentované možnosti telefonu, by bylo v budoucnu třeba hlasové ovádání rozšířit.

6.2.4 Předtestový dotazník

Předtestovým dotazníkem (viz. příloha F) se ověřil rozsah implementovaných schopností hlasového rozpoznávání, který se zásadně nelišil od implementovaných shopností (viz. obrázek 6.2), jejichž podoba byla získána od druhé skupiny testujících uživatelů a využita v kapitole 5.1.

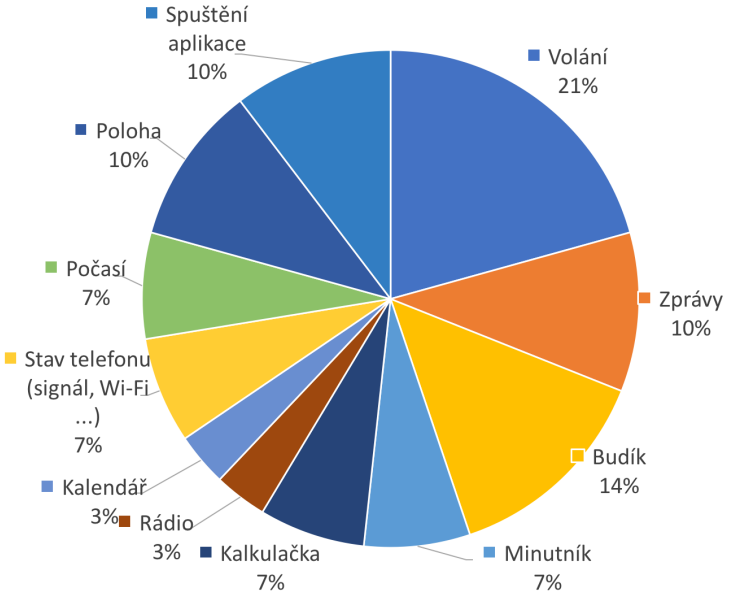


Obrázek 6.2. Statistika uživateli žádaných aplikací z předtestového dotazníku

Uživatelé, kteří dotazník vyplnili, do jednoho uvedli (viz. příloha F), že si umí představit, že by běžné funkce telefonu ovládali právě hlasem. Dále dotazník zjistil, že ovládání telefonu hlasem na veřejnosti už všichni uživatelé za možné nepovažují nebo, že rozhodně nebudou používat na veřejnosti všechny funkce jako například diktování textu zpráv.

6.2.5 Potestový dotazník

Všichni uživatelé, kteří vyplnili potestový dotazník, uvedli (viz. příloha G), že aplikaci testovali déle než 2 hodiny (viz. příloha G). Uspěšnost správného pochopení a vykonání požadovaného příkazu byla dle uživatelů průměrně 77,5%. Z dat ukládaných do databáze vyplývá, že v tomto shrnutí budou zahrnuty i pokusy o vykonání požadavku, který zatím nebyl implementován.



Obrázek 6.3. Statistika uživateli nejvíce užitečných aplikací z potestového dotazníku

Hlasový asistent byl také představen na letošní největší výstavě pomůcek pro nevidomé a slabozraké na světě, SightCity [1], která se konala v období od 3.5 do 5.5.2017 v německém městě Frankfurtu. Během této výstavy byly testovánny německé a anglické jazykové mutace hlasového asistenta a setkaly se taktéž s pozitivními ohlasy.

[1] `http://www.sightcity.net`

Kapitola 7
Další rozvoj projektu

Projekt, popsaný v této práci, není sepsáním této práce definitivně uzavřen, ale bude se na jeho vývoji nadále pokračovat. Tato kapitola popisuje předpokládaný směr, jímž se v nejbližší době, řekněme jednoho roku, bude práce na tomto projektu ubírat.

7.1 Rozšíření podporovaných jazyků

Prvním bodem, na kterém se bude dále pracovat, bude rozšíření jazykové podpory. Pořadí implementace podpory dalších jazyků bude záviset na poptávce z dané oblasti a schopnosti získat testovací skupinu betatestrů, kteří implementaci v daném jazyce vyzkouší.

Pro fungování v daném jazyce je třeba reálně pouze vyplnit, přeložit, sadu klíčových slov (viz. kapitola 5.2). Abychom ovšem mohli ověřit alespoň základní funkčnost před samotným započetím beta testování, bude spíše důležité přeložit množinu testovacích vět. Žádoucí také bude překladateli, alespoň principielně, vysvětlit způsob fungování, aby testovací set vět nejenom překládal, ale zároveň také doplňoval o různé varianty příkazů v daném jazyce, které by měly vést ke stejnému výsledku hlasového asistenta.

V současnosti je již rozhodnuto, že dalším podporovaným jazykem bude francouzština, neboť splňuje výše uvedené podmínky. Na základě zkušeností z této implementace budou následně přidávány další jazyky.

7.2 Zavedení zpětné vazby mezi každodenní uživatele

Aby bylo možné projekt nadále rozšiřovat a reagovat na potřeby uživatelů, bude třeba zavést rozsáhlejší zpětnou vazbu, než tu, která byla popsaná v kapitole 6.1. Její hlavní nevýhoda spočívá v tom, že její právní legalita je podmíněna písemným souhlasem uživatele. Modernějším pojetím by bylo rozhodnutí o povolení ukládání provozních dat hlasového asistenta, získávat přímo z mobilního telefonu. V tomto směru bude ovšem třeba vyřešit právní stránku takového ukládání údajů. Právní podmínky se navíc mohou, a pravděpodobně i budou, v jednotlivých zemích lišit.

7.3 Využití strojového učení pro pochopení významu vět

Jak bylo popsáno v několika předchozích kapitolách (viz kapitoly 4 a 5), způsob použitý v této práci pro pochopení významu uživatelských příkazů je plně založen na slovníku klíčových slov pro jednotlivé podporované operace. Výhodou tohoto přístupu je deterministické chování. Nevýhodou ovšem je, že jakékoli rozšíření schopností je třeba

programově implementovat. Oddělit od sebe dva potencionálně možné výsledky je nakonec možné pouze soustavou prahových hodnot a stavbou slovníku.

V tomto směru by bylo vhodné zavést možnost dnes tak moderního strojového učení, a umožnit tak rozpoznávacímu systému reagovat více flexibilně například na možné změny chování Google Cloud to Speech API a podobně. Tuto kapitolu nebude ovšem možné realizovat dříve, než se podaří zavézt zpětnou vazbu od běžných uživatelů, jak je popsáno v předchozí kapitole (viz. kapitola 7.2). K vytvoření fungující neuronové sítě bude totiž třeba velkého množství korektně ohodnocených dat. Část máme již k dizpozici právě díky tomuto projektu, ovšem pro potřeby vycvičení neuronové sítě není toto množství zdaleka dostačující.

Problémem bude pro neuronovou síť například vyhledávání kontaktů ve větě. Současná implementace má tu výhodu, že ve skutečnosti v příkazu hledá jen ty kontakty, které jsou v uživatelově adresáři. Tento přístup bude ovšem v prostředí neuronové sítě velmi obtížně reprodukovatelný. Stejně tak bude problém ve vyhledávání rádiových stanic či hudby. I tato implementace je aktuálně závislá na dostupném seznamu.

7.4 Tensorflow

Tensorflow[1] je softwarová knihovna od Googlu pro výpočty metody data flow graph, kterou v listopadu roku 2015 uvolnil Google jako open source. Právě díky této knihovně mouhou aplikace získat schopnost se učit [20]. Od jeho uvedení se na jeho základě vytvořila spousta aplikací s rozdílným účelem, které jsou dostupné na GitHubu.

Mezi těmito projekty nalezneme i takové, věnující se právě oblasti nature language programing, které by mohly být využity.

7.4.1 SyntaxNet

Jako příklad můžeme uvést projekt SyntaxNet [22] [2], jehož cílem je zjistit slovní druhy všech slov ve větě a určit větné členy. Využitím této infomace v již popsaném algoritmu bychom mohli například zmenšit prohledávanou oblast věty jen na určité slovní druhy či větné členy. Nebo zvýšit důležitost shody se slovníkem právě ve chvili, kdy se jedná o určitý slovní druh nebo větný člen a podobně.

Ve snaze využití tohoto přístupu byly již podniknuty první kroky, které zahrnují servrovou implementaci projektu SyntaxNet a vytvoření jejího REST API pro použití z telefonu. Vytvořené API dovoluje na server poslat dotaz s určením jazyka modelu, který má SyntaxNet použít. Aktuálně podporované jazyky jsou shodné s jazyky podporovanými během této práce, tedy čeština, angličtina a němčina.

Důvody, které znemožnily použití tohoto přístupu ve finální implementaci byly v zásadě dva. Prvním z nich byla skutečnost, že SyntaxNet předpokládá, že dostane na vstupu text, kde každá věta je, tak jak je zvykem, ukončena tečkou. Jestliže ve vstupu tečka není, předpokládá se, že se jedná o jednu větu. Bohužel Google Cloud Speech to Text API (viz. kapitola 3.2.1), které se v této práci používá pro převod řeči do textové podoby, touto schopností od začátku roku 2017 nedisponuje.

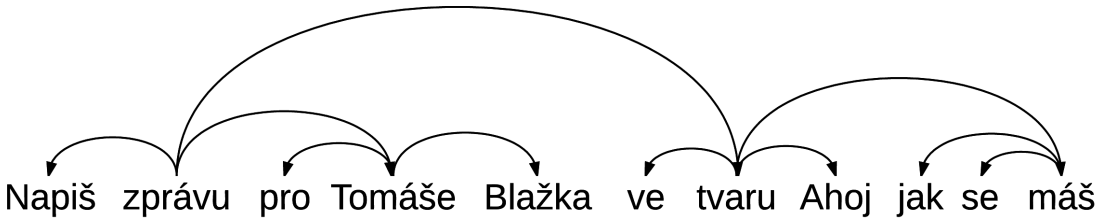
Druhým zásadním problémem je výpočetní náročnost provozu SyntaxNet. Na serveru na kterém byla implementace zprovozněna, trval překlad věty řádově několik vteřin, což je poměrně hodně. Tensorflow má podporu pro využití vysokého výpočetního výkonu

[1] `https://www.tensorflow.org`
[2] `https://github.com/tensorflow/models/tree/master/syntaxnet`

CUDA jader grafických karet, které by tento neduh mohlo minimálně částečně eliminovat. V tomto směru vyvstává ovšem další otázka, týkající se provozních nákladů takového servrového řešení.

Představu o funkčnosti a výstupu SyntaxNetu může dát následující příklad. Obrázek 7.1 ukazuje grafické znázornění získaných větných závislostí.

```
1 Napiš    _ C=------------ C=------------ _ 2 nummod  _ _
2 zprávu   _ NNFS4-----A---- NNFS4-----A---- _ 0 ROOT    _ _
3 pro      _ RR--4---------- RR--4---------- _ 4 case    _ _
4 Tomáše   _ NNMS4-----A---- NNMS4-----A---- _ 2 nmod    _ _
5 Blažka   _ NNMS4-----A---- NNMS4-----A---- _ 4 name    _ _
6 ve       _ RV--6---------- RV--6---------- _ 7 case    _ _
7 tvaru    _ NNIS6-----A---- NNIS6-----A---- _ 2 dep     _ _
8 Ahoj     _ C=------------ C=------------ _ 7 nummod  _ _
9 jak      _ Db------------ Db------------ _ 11 mark   _ _
10 se      _ P7-X4---------- P7-X4---------- _ 11 expl   _ _
11 máš     _ C=------------ C=------------ _ 7 nummod  _ _

Time: 7 s
```



Obrázek 7.1. Grafické znázornění výstupu SyntaxNetu

7.4.2 Využití CNN pro klasifikaci vět

Další zamýšlenou možností využití neuronové sítě Tensorflow je implementace CNN (konvoluční neuronové sítě), která se velmi často používá například na klasifikaci obrázků. Tedy například na to, zda obrázek nějaký předmět obsahuje nebo neobsahuje.

Přístup popsaný v [1] ukazuje, že CNN se s velmi dobrými výsledky dá použít i na klasifikaci textu. V tomto místě ovšem již budeme potřebovat opravdu velké množiny dat, které musí obsáhnout celou paletu případů, které chceme klasifikovat a které současná implementace klasifikuje na základě klíčových slov a jejich synonym.

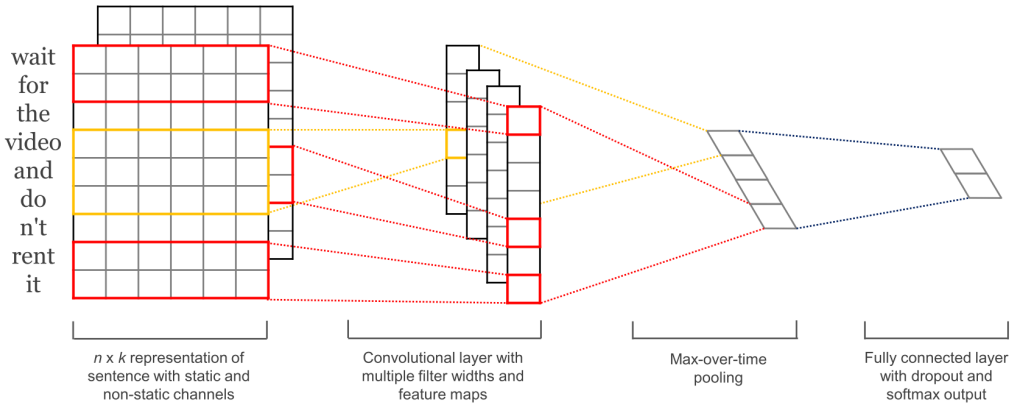
Na GitHubu[1] se dá opět najít implementace [1] využívající právě Tensorflow. Tato implementace ovšem obsahuje data, rozdělená pouze na množiny pozitivních a negativních vět. Úkolem poté je klasifikovat věty právě mezi těmito dvěma stavy. V [1] je ovšem popsaná klasifikace mezi několika množinami, například mezi větami velmi pozitivními, pozitivními, neutrálními, negativními a velmi negativními. Další uvedená množina dat rozhoduje o tom, zda se otázka týká člověka, polohy, výpočtu a podobně. Popsaný přístup pravděpodobně nebude vhodný ke kompletnímu nahrazení algoritmu, popsanému v této práci. S dostatečně kvalitní a velkou trénovací množinou by ovšem mohl doplnit nebo dokonce nahradit první část popsaného vyhledávacího algoritmu (viz. kapitola 5.4.1) a rozhodovat, které aplikace se daný příkaz týká.

[1] https://github.com/dennybritz/cnn-text-classification-tf

Základní myšlenka, popsaná v [1], je taková, že pomocí konvoluce se hledá shoda aktuálně zadané věty, respektive jejích částí, s větami, které jsou uloženy a klasifikovány ve slovníku.

Slovník vzniká z trénovacích dat, kde každému slovu je přiřazeno číslo. Každá věta je tedy reprezentována jako vektor.

Věta zadaná ke klasifikaci je vyjádřena vektorem, tedy stejným způsobem jako věty z trénovacího slovníku. Následně je tento vektor konvolucí porovnáván se slovníkem. Během tohoto porovnávání je využito několika filtrů jako klouzání přes 3,4 a 5 slov najednou. Dále výsledek principem max-pooling [35] převede na jeden dlouhý vektor, z kterého posléze určí výsledek (viz. obrázek 7.2).



Obrázek 7.2. Model architektury CNN pro klasifikaci věty (převzato z [1])

Před samotnou implementací bude třeba sestavit trénovací množiny dat, kde každá množina bude obsahovat řádově tisíce vět obdobného významu. Za vyzkoušení bude jistě stát například předprohledání vět současným způsobem se snahou najít v nich kontakt a ten následně nahradit nějakými zástupnými znaky, aby se eliminovala možnost, že v trénovací množině dat nebude právě toto jméno.

Dalším možným zásahem do algoritmu by mohla být změna porovnávání dvou slov ve stylu popsaném v kapitole 4.2.

Kapitola 8
Závěr

Cílem práce bylo prostudovat problematiku ovládání mobilního telefonu hlasem s ohledem na nevidomé uživatele a následně navrhnout systém, který by ovládání mobilního telefonu hlasem umožňoval. Implementace měla být provedena na operačním systému Android a následně otestována na alespoň pěti uživatelích.

Před započetím vývoje vlastní aplikace jsem zevrubně testoval a analyzoval dnes dostupná řešení hlasových asistentů v mobilních telefonech napříč platformami.

Aplikace, vytvořená v rámci této práce, se skládá ze tří samostatných částí. První část aplikace se stará o převod mluveného slova do textové podoby, druhá část aplikace řeší pochopení významu uživatelem vyřčeného příkazu a třetí část aplikace požadovaný příkaz provádí.

Tato hierarchie dává do budoucna možnost jednotlivé části aplikace samostatně rozvíjet či reimplementovat s využitím jiného přístupu než toho použitého v této práci a následně je opět začlenit zpět do fungujícího celku.

Uživatelské rozhraní aplikace je navrženo s důrazem na jednoduchost ovládání a je vhodné pro nevidomé a slabozraké uživatele s kterými bylo, v rámci organizace SONS (Sjednocená organizace nevidomých a slabozrakých ČR), diskutováno. Kromě ovládání vzešlo z této diskuze mnoho dalších cenných připomínek, týkajících se chování a návyků nevidomých a slabozrakých uživatelů. Každý provedený krok je tedy ozvučen a ovládání samotné je založeno na několika málo gestech.

Aplikace byla mimo jiné otestována na konferenci SightCity 2017, kde vyvolala pozornost mezi návštěvníky. Během testování na reálných potenciálních uživatelích byla získána pozitivní zpětná vazba na implementované schopnosti, stejně jako cenné návrhy na další rozšíření a vylepšení.

Za hlavní vyhodu svého řešení považuji deterministické chování výsledné aplikace a skutečnost, že její vývoj od počátku počítal s fungováním v českém jazyce, který je lingvisticky jedním z nejsložitějších. Aplikace umožňuje nevidomým uživatelům po krátké instruktáži zjednodušit a zefektivnit používání jejich mobilního telefonu.

Jako vedlejší produkt této práce vznikla ohodnocená databáze obsahující příkazy a dotazy získané přímo od uživatelů. Tato databáze je přílohou této práce a obsahuje data v českém, anglickém a německém jazyce a může být použita při další projektech zabívajících se nature language programing.

Další rozvoj aplikace by se měl ubírat směrem k podpoře více jazyků a zapojení strojového učení na základě získaných dat z testování i následného nasazení aplikace.

Literatura

[1] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *CoRR*. 2014, abs/1408.5882

[2] V. Gaudissart, S. Ferreira, C. Thillou a B. Gosselin. *Sypole: A mobile assistant for the blind*. In: *2005 13th European Signal Processing Conference*. 2005. 1-4. http://ieeexplore.ieee.org/document/7078510/.

[3] Shiri Azenkot a Nicole B. Lee. *Exploring the Use of Speech Input by Blind People on Mobile Devices*. In: *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: ACM, 2013. 11:1–11:8. ISBN 978-1-4503-2405-2. http://doi.acm.org/10.1145/2513383.2513440.

[4] Shiri Azenkot. *Eyes-Free Input on Mobile Devices*. Disertační práce, University of Washington. 2014.

[5] TIMOTHY TORRES. *Google Home vs. Amazon Echo: Which One Should Rule Your Smart Home?* http://www.pcmag.com.

[6] Judy Sanhz. *Here is some best Alexa commands for your Amazon Echo*. https://knowtechie.com/the-best-alexa-commands-for-amazon-echo/.

[7] Google. *Google Home*. https://madeby.google.com/home/.

[8]

[9] P. Roach. *English Phonetics and Phonology: A Practical Course*. Cambridge University Press, 2000. ISBN 9780521786133. https://books.google.cz/books?id=u29ff2oIPk8C.

[10] Krčmová Marie. *Fonetika a fonologie*. https://is.muni.cz/do/rect/el/estud/ff/js08/fonetika/ucebnice/ch08s01s01.html.

[11] Dong Yu a Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014. ISBN 1447157788, 9781447157786.

[12] Ed Grabianowski. How Speech Recognition Works, John Garofolo. *How Stuff Works*. 2016, http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition.htm.

[13] Leonard E. Baum a Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Ann. Math. Statist.*. 1966, 37 (6), 1554–1563. DOI 10.1214/aoms/1177699147.

[14] *Android developer*. http://developer.android.com.

[15] *Microsoft Bing Speech API: Android Speech-to-Text Client Library & Sample*. https://github.com/Microsoft/Cognitive-Speech-STT-Android.

[16] *Bing Speech API*.
https://www.microsoft.com/cognitive-services/en-us/speech-api/documentation/overview.

[17] H. A. Maarif, R. Akmeliawati, Z. Z. Htike a T. S. Gunawan. *Complexity Algorithm Analysis for Edit Distance.* In: *2014 International Conference on Computer and Communication Engineering*. 2014. 135-137.

[18] Bo Chen Le Sun Xianpei Han a Bo An. Sentence rewriting for semantic parsing.

[19] Hoifung Poon a Pedro Domingos. *Unsupervised semantic parsing.* In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. 2009. 1–10.

[20] David MW Powers a Christopher CR Turk. *Machine learning of natural language*. Springer Science & Business Media, 2012.

[21] M. A. Tayal, M. M. Raghuwanshi a L. Malik. *Syntax Parsing: Implementation Using Grammar-Rules for English Language.* In: *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*. 2014. 376-381.

[22] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov a Michael Collins. Globally Normalized Transition-Based Neural Networks. *CoRR*. 2016, abs/1603.06042

[23] Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning a Christopher Potts. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*. 2016,

[24] "Stavros Konstantinidis". "Computing the edit distance of a regular language". *"Information and Computation"*. "2007", "205" ("9"), "1307 - 1316". DOI "http://dx.doi.org/10.1016/j.ic.2007.06.001". "".

[25] "Giovanni Pighizzini". "How Hard Is Computing the Edit Distance?". *"Information and Computation"*. "2001", "165" ("1"), "1 - 13". DOI "http://dx.doi.org/10.1006/inco.2000.2914". "".

[26] MEHRYAR MOHRI. EDIT-DISTANCE OF WEIGHTED AUTOMATA: GENERAL DEFINITIONS AND ALGORITHMS. *International Journal of Foundations of Computer Science*. 2003, 14 (06), 957-982. DOI 10.1142/S0129054103002114.

[27] Sandeep Hosangadi. Distance measures for sequences. *arXiv preprint arXiv:1208.5713*. 2012,

[28] L. Yujian a L. Bo. A Normalized Levenshtein Distance Metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2007, 29 (6), 1091-1095. DOI 10.1109/TPAMI.2007.1078.

[29] Alexandr Andoni a Krzysztof Onak. Approximating Edit Distance in Near-Linear Time. *CoRR*. 2011, abs/1109.5635

[30] L Kari, S KONSTANTlNIDlS, S Perron, G WOZNIAKI a J KU. Computing the Hamming Distance of a Regular l Language in Quadratic Tirne˜.

[31] World Heritage Encyclopedia. *DAMERAU–LEVENSHTEIN DISTANCE*.
http://self.gutenberg.org/articles/eng/Damerau{Levenshtein_distance.

[32] Naoko Miura Tomohiro Takagi. WSL: Sentence Similarity Using Semantic Distance Between Words. *SemEval-2015*. 2015, 128.

[33] *Sjednocená organizace nevidomých a labozrakých ČR*.
http://www.sons.cz.

[34] Bc. Petr Svobodník. *Zpřístupnění mobilních telefonů se systémem Android pro nevidomé uživatele*. 2013.
http://www.diplomovaprace.cz/2013/54/Svobodnik-thesis-2013.pdf.

[35] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber a L. M. Gambardella. *Max-pooling convolutional neural networks for vision-based hand gesture recognition.* In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. 2011. 342-347.

Příloha A
Zkratky a symboly

A.1 Zkratky

Seznam zkratek použitých v práci.

GUI	Graphical User Interface - grafické uživatelské rozhraní
API	Application Programming Interface
JAVA	objektově orientovaný programovací jazyk, též americký slangový výraz pro kávu po němž má tento jazyk svůj název
SONS	Sjednocená organizace nevidomých a slabozrakých České republiky
ADT	Android Developer Tools
SDK	Software Developer Kit
TTS	Text to speech, převod textu na mluvené slovo
DARPA	Defense Advanced Research Projects Agency je agentura amerického ministerstva obrany, která je zodpovědná za vývoj nových vojenských technologií
ROC	Receiver Operating Characteristic
CNN	Konvoluční neuronová síť
MySQL	systém řízení báze dat uplatňující relační databázový model
REST	
GSM	globální systém mobilní komunikace

A.2 Symboly

s	sekundy

Příloha B
Obsah přiloženého CD

/Diplomová práce	Tato práce ve formátu PDF
/Obrázky	Obrázky použité v této práci
/Program	Zdrojové kódy programu
/Program/Android aplikace	Zdrojový kód Android části aplikace
/Program/BlindVoiceDecoder	Zdrojové kódy programu BlindVoiceDecoder.jar sloužícího k prohledávání věty a nalezení jejího významu
/Přílohy	Přílohy práce
/Trénovací data	Ohodnocená databáze získaných dat použitých k trénování aplikace

Příloha C
Sběr testovacích dat

C.1 Ovládání hlasového asistenta

Cílem dotazníku je získat data o tom jakým způsobem lidé mluví na umělou inteligenci v očekávání splnění příkazu. Umělou inteligencí v tomto dotazníku bude mobilní telefon se zapnutým hlasovým asistentem. Představete si například Siri od Applu, Google now nebo Cortanu. Tímto směrem budou i tématicky zaměřené úkoly.

Předem děkuji za váš čas, Michal Blažek

C.1.1 Volání

Vytáčení kontaktů a číslel s pomocí hlasového asistenta

1. Napište co by jste řekli telefonu, kdyby jste chtěli, aby zavolal vaší mamince.
2. Napište, co by jste řekli telefonu, kdyby jste chtěli, aby zavolal vašemu známého Jiřího Nováka.
3. Napište, co by jste řekli telefonu, kdyby jste chtěli aby vytočil číslo 775123456

C.1.2 Psaní zpráv

Psaní zpráv s pomocí hlasového asistenta

1. Napište co byste řekli telefonu, kdybyste chtěli, aby poslal mamince textovou zprávu o tom jak se máte na dovolene (text si vymyslete).
2. Napište co byste řekli telefonu, kdybyste chtěli, aby poslal Tomášovi textovou zprávu s tím co má nakoupit k večeři (opět se fantazii meze nekladou).
3. Napište co byste řekli telefonu, kdybyste chtěli, aby poslal kamarádce/kamarádovi textovou zprávu, že dorazíte pozdě
4. Vymyslete si další obdobný scénář prosím
5. A ještě jeden prosím

C.1.3 Ovládání budíku

Ovládání budíku s pomocí hlasového asitenta

1. Napište co byste řekli telefonu, kdybyste chtěli, aby vytvořil budík na 8 hodin ráno
2. Napište co byste řekli telefonu, kdybyste chtěli, aby vytvořil budík na 16:30 (nebo půl páté odpoledne)

C.1.4 Ovládání časovače

Ovládání odpočtu s pomocí hlasového asistenta

1. Napište co byste řekli telefonu, kdybyste chtěli, aby nastavil odpočet na 6 minut
2. Napište co byste řekli telefonu, kdybyste chtěli, aby nastavil odpočet na 1 hodinu a 30 minut

C.1.5 Předpověď počasí

Předpověď počasí s pomocí hlasového asistenta

1. Napište co byste řekli telefonu, kdybyste chtěli, aby Vám řekl jaké bude zítra počasí.
2. Napište co byste řekli telefonu, kdybyste chtěli, aby Vám řekl jaké bude počasí v pátek (např. že je to 23.6.2017)

C.1.6 Ovládání kalendáře

Přidávání úkolů do kalendáře

1. Napište co byste řekli telefonu, kdybyste chtěli, aby telefon vytvořil událost na 23.6.2017 na 11 hodiny dopoledne
2. Napište co byste řekli telefonu, kdybyste chtěli, aby vytvořil událost na příští úterý na 4 hodiny odpoledne.

C.1.7 Cokoli dalšího

Napište prosím cokoliv dalšího, co Vám v dotazníku chybělo a chtěli by jste, aby to hlasový asistent uměl. Uveďte prosím jak příkaz tak konkrétní akci, kterou by měl telefon udělat.

Případně napište cokoliv dalšího co Vás napadá, že by jste chtěli autorovi sdělit.

Příloha D
Předtestovací dotazník

D.1 Obecný průzkum

Tento dotazník je určený pro nevidomé uživatele mobilního dotykového telefonu Blindshell. V první části dotazníku prosím vyplňte několik málo informací o sobě a Vašem postižení. Druhá část dotazníku se snaží zjistit co a jak by mě hlasový asistent v mobilním telefonu dělat/nedělat tak aby nevidomému člověku co možná nejvíce pomohl s ovládáním každodenně používaných funkcí v telefonu.

1. Kolik je Vám let ? (vyberte jednu možnost)
   - 0 - 20
   - 21 - 40
   - 41 - 50
   - 51 a výše
2. Která z uvedených možností odpovídá Vašemu handicapu ? (vyberte jednu nebo více možností)
   - úplná nebo praktická slepota
   - zbytky zraku
   - poruchy binokulárního/barevného vidění
   - slabozrakost
   - postižené obě oči
   - porucha zraku je trvalá
   - porucha zraku je krátkodobá nebo opakující se
3. Jak dlouho máte handicap ? (vyberte jednu možnost)
   - 1 rok a více
   - 5 a více let
   - 10 a více let
   - od narození

D.2 Hlasový asistent

1. Umíte si představit že ovládáte běžné funkce telefonu (volání, psaní zpráv, nastavení budíku atd.) hlasovými příkazy? (vyberte jednu možnost)
   - ano
   - ne

2. Ovládali by jste telefon hlasem i na veřejnosti? hlasovými příkazy? (vyberte jednu možnost)

   - ano
   - ne

3. Vyberte prosím funkce telefonu, jejichž ovládaní hlasem by se Vám hodilo nejvíce a ušetřilo by Vám nejvíce času. (vyberte jednu nebo více možností)

   - vytáčení kontaktů a čísel - [Př.: „Zavolej Petra Nováka“]
   - psaní sms - [Př.: „Napiš zprávu budu mít 10 minut spoždění pro Petra Nováka“]
   - psaní emailů - [Př.: „Napiš email pro Petra Nováka s textem schůzka by se mi hodila v pondělí“]
   - ovládání budíku - [Př.: „Nastav budík na 7 hodin a 30 minut dopoledne“]
   - ovládání časovače a stopek - [Př.: „Spusť odpočet na 5 minut“]
   - ovládání kalendáře (vytváření událostí, úkolů atd.) - [Př.: „Vytvoř mi úkol na 16. dubna v 11h“]
   - diktování výpočtů (krátké každodenní výpočty) - [Př.: „Kolik je 17
   - 4?“]
   - zjištění předpovědi počasí na daný den - [Př.: „Jaké bude zítra počasí?“]
   - zjištění aktuálního stavu telefonu (hodiny, datum, stav baterie, signálu ...)
   - nastavení funkcí telefonu (zapnout/vypnout WiFi, mobilní data, vyzvánění ...)
   - ovládání hudeního přehrávače nebo rádia [Př.: „Zapni evropu 2“, „Pusť Green day“]
   - vyhledávání kontaktů [Př.: „Najdi kontakt Petr Novotný“]
   - ovládání knihovny [Př.: „Otevři poslední knížku“]
   - zjištění aktuální polohy [Př.: „Kde to jsem?“, „Co je to za ulici?“]

Příloha E
Potestovací dotazník

E.1 Obecný průzkum

Tento dotazník je určený pro nevidomé uživatele mobilního dotykového telefonu Blindshell. V první části dotazníku prosím vyplňte několik málo informací o sobě a Vašem postižení. Druhá část dotazníku je určena uživatelům, kteří již měli možnost otestovat hlasové ovládání telefonu pro nevidomé Blindshell vyzkoušet a mají tímto možnost poskytnout zpětnou vazbu pro zjištění jejich spokojenosti.

1. Kolik je Vám let ? (vyberte jednu možnost)
   - 0 - 20
   - 21 - 40
   - 41 - 50
   - 51 a výše
2. Která z uvedených možností odpovídá Vašemu handicapu ? (vyberte jednu nebo více možností)
   - úplná nebo praktická slepota
   - zbytky zraku
   - poruchy binokulárního/barevného vidění
   - slabozrakost
   - postižené obě oči
   - porucha zraku je trvalá
   - porucha zraku je krátkodobá nebo opakující se
3. Jak dlouho máte handicap ? (vyberte jednu možnost)
   - 1 rok a více
   - 5 a více let
   - 10 a více let
   - od narození

E.2 Hlasový asistent

1. Jak dlouho jste hlasového asistenta testovali?
   - 0 až 1 hodinu
   - 1 až 2 hodinu
   - 2 až 4 hodinu
   - více

2. Označte prosím 5 dle Vás nejužitečnějších funkcí se kterými Vám hlasový asistent pomohl

   - Vytáčení kontaktů
   - Psaní SMS
   - Ovládání budíku
   - Ovládání časovače
   - Ovládání kalendáře
   - Zjišťování předpovědí počasí
   - Matematické výpočty
   - Aktuální poloha
   - Zjištění aktuálního stavu telefonu
   - Spuštění aplikace telefonu obecně
   - Spuštění rádiové stanice
   - Jiné

3. S jakou úspěšností (v procentech) dle Vás plnil hlasový asistent Vaše požadavky.

   - 0 %
   - 10 %
   - 20 %
   - 30 %
   - 40 %
   - 40 %
   - 50 %
   - 60 %
   - 70 %
   - 80 %
   - 90 %
   - 100 %

Příloha F
Odpovědi na předtestový dotazník

Uživatel	Kolik je vám let?	Která z uvedených možností odpovídá vašemu handikepu?	Jak dlouho máte handikep?	Jak často používáte mobilní telefon?
1	21-40	def	5 a více	několikrát deně
2	21-40	de	5 a více	několikrát deně
3	21-40	h	od narození	několikrát deně
4	21-40	h	od narození	několikrát deně
5	21-40	bdef	od narození	několikrát deně
6	21-40	aef	od narození	několikrát deně

Tabulka F.1. Odpovědi na předtestový dotazník - Obecný průzkum

Uživatel	Umíte si představit, že ovládáte běžné funkce telefonu hlasovými příkazi	Ovládali by jste telefon hlasem i na veřejnosti?	Vyberte funkce, které by se vám nejvíce hodilo ovládat hlasem
1	ano	ano	abcgh(ch)jkl
2	ano	ano	abchj
3	ano	ano	abfl
4	ano	ano	bf
5	ano	ano	adeh(ch)i
6	ano	ne	abdefgh(ch)ijkl

Tabulka F.2. Odpovědi na předtestový dotazník - Hlasový asistent

Příloha G
Odpovědi na potestový dotazník

Uživatel	Kolik je vám let?	Která z uvedených možností odpovídá vašemu handikepu?	Jak dlouho máte handikep?	Jak často používáte mobilní telefon?
1	21-40	def	5 a více	několikrát deně
2	21-40	de	5 a více	několikrát deně
3	21-40	h	od narození	několikrát deně
4	21-40	h	od narození	několikrát deně
5	21-40	bdef	od narození	několikrát deně
6	21-40	aef	od narození	několikrát deně

Tabulka G.3. Odpovědi na potestový dotazník - Obecný průzkum

Uživatel	Jak dlouho jste hlasového asistenta testovali?	Označte 5 nejužitečnějších funkcí se kterými Vám hlasový asistent pomohl?	S jakou úspěšností dle Vás plnil hasový asistent Vaše požadavky?
1	více než 4 h	agh(ch)	80 %
2	2 až 4 h	abcdf	70 %
3	2 až 4 h	abceh	80 %
4	více než 4 h	abhij	80 %
5	více než 4 h	acf(ch)i	70 %
6	2 až 4 h	acdgi	80 %

Tabulka G.4. Odpovědi na potestový dotazník - Hlasový asistent

Příloha H
Trénovací data pro dekódování smyslu textu

H.1 Česká trénovací data

```
<resources>
    <sentence id="0">
        <input>Napiš zprávu pro Tomáše Bartáka</input>
        <control>[APP]=Message;[CONTACT]=[Barták Tomáš];[TEXT]=null
        </control>
    </sentence>
    <sentence id="1">
        <input>Zavolej Tomáše Blažka</input>
        <control>[APP]=Call;[CONTACT]=[Blažek Tomáš]</control>
    </sentence>
    <sentence id="2">
        <input>Zavolej Grofovou</input>
        <control>[APP]=Call;[CONTACT]=[Grofová Monča, Grofová Radka, ...]
        </control>
    </sentence>
    <sentence id="3">
        <input>Vytoč Tomáše</input>
        <control>[APP]=Call;[CONTACT]=[Blažek Tomáš, Barták Tomáš, ...]
        </control>
    </sentence>
    <sentence id="4">
        <input>Zavolej Blažkovi</input>
        <control>[APP]=Call;[CONTACT]=[Blažek Tomáš]</control>
    </sentence>
    <sentence id="5">
        <input>Napiš zprávu pro Tomáše Blažka s textem Hrozně se
        těším na dovolenou</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=Hrozně se těším na dovolenou</control>
    </sentence>
    <sentence id="6">
        <input>Napiš zprávu pro Tomáše Blažka kde text je Hrozně se
        těším na dovolenou</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=Hrozně se těším na dovolenou</control>
    </sentence>
    <sentence id="7">
        <input>Napiš zprávu s textem Kup kopu vajec a chilli pro
        Tomáše Bartáka</input>
        <control>[APP]=Message;[CONTACT]=[Barták Tomáš];
```

```
        [TEXT]=Kup kopu vajec a chilli</control>
    </sentence>
    <sentence id="8">
        <input>Napiš zprávu Kup kopu vajec a chilli pro
        Tomáše Bartáka</input>
        <control>[APP]=Message;[CONTACT]=[Barták Tomáš];
        [TEXT]=Kup kopu vajec a chilli</control>
    </sentence>
    <sentence id="9">
        <input>Napiš zprávu Kup kopu vajec a chilli pro Bartáka Tomáše
        </input>
        <control>[APP]=Message;[CONTACT]=[Barták Tomáš];
        [TEXT]=Kup kopu vajec a chilli</control>
    </sentence>
    <sentence id="10">
        <input>Napiš zprávu Blažek Tomáš Kup kopu vajec a chilli</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=Kup kopu vajec a chilli</control>
    </sentence>
    <sentence id="11">
        <input>Napiš zprávu pro Blažek Tomáš Kup kopu vajec pro tátu
        </input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=Kup kopu vajec pro tátu</control>
    </sentence>
    <sentence id="12">
        <input>Napiš zprávu pro Blažek Tomáš s textem Kup kopu vajec pro
        tátu</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=Kup kopu vajec pro tátu</control>
    </sentence>
    <sentence id="13">
        <input>Napiš zprávu s textem Kup kopu vajec pro tátu pro Blažka
        Tomáše </input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=Kup kopu vajec pro tátu</control>
    </sentence>
    <sentence id="14">
        <input>Napiš Jirkovy zprávu jsem v tramvaji, budu tam za 15min
        </input>
        <control>[APP]=Message;[CONTACT]=[Petráň Jiří, ...];
        [TEXT]=jsem v tramvaji, budu tam za 15min</control>
    </sentence>
    <sentence id="15">
        <input>Napiš zprávu Tomášovi</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš, Barták Tomáš
        , ...];
        [TEXT]=null</control>
    </sentence>
    <sentence id="16">
        <input>Napiš zprávu Janu Kapounovou ve tvaru Posílám pozdrav z
        Itálie máme se tu fajn svítí sluníčko a je teplo</input>
        <control>[APP]=Message;[CONTACT]=[Kapounová Janďa];
```

```
        [TEXT]=Posílám pozdrav z Itálie máme se tu fajn svítí sluníčko a
        je teplo</control>
    </sentence>
    <sentence id="17">
        <input>Napiš zprávu Tomášovi Mládkovi ve tvaru kup nějakou
        rybu a mraženou zeleninu udělám z toho pizzu.</input>
        <control>[APP]=Message;[CONTACT]=[Mládek Tomáš];
        [TEXT]=kup nějakou rybu a mraženou zeleninu udělám z toho pizzu.
        </control>
    </sentence>
    <sentence id="18">
        <input>Odešli SMS Tomášovi Čumpelíkovi zdar tome, nakup ňáký
        lahváče, budu dělat vošouchy, tak kup ještě brambůry a majoránku.
        A makej!</input>
        <control>[APP]=Message;[CONTACT]=[Čumpelík Tomáš];
        [TEXT]=zdar tome, nakup ňáký lahváče, budu dělat vošouchy,
        tak kup ještě brambůry a majoránku. A makej!</control>
    </sentence>
    <sentence id="19">
        <input>Napiš Tomášovi Mládokovi čau, kup k večeři půl kila
        mletého hovězího, špagety, dvě mrkve, celer, tři petržele,
        loupaná rajčata, máslo a bílé víno.
         To co mám rád. A nezapomeň na dvacet deka šunky. Díky.</input>
        <control>[APP]=Message;[CONTACT]=[Mládek Tomáš];
        [TEXT]=čau, kup k večeři půl kila mletého hovězího, špagety,
        dvě mrkve, celer, tři petržele, loupaná rajčata, máslo a bílé
        víno. To co mám rád. A nezapomeň na dvacet deka šunky. Díky.
        </control>
    </sentence>
    <sentence id="20">
        <input>Pošli Tomášovi Blažkovi zprávu že má koupit 3 vejce,
        mléko, špagety a nějaké pečivo</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=že má koupit 3 vejce, mléko, špagety a nějaké pečivo
        </control>
    </sentence>
    <sentence id="21">
        <input>Napiš Jirkovi Petráňovi zprávu na pivo přijdu pozdě
        </input>
        <control>[APP]=Message;[CONTACT]=[Petráň Jiří];
        [TEXT]=na pivo přijdu pozdě</control>
    </sentence>
    <sentence id="22">
        <input>Napiš Aničce Tatíčkové promiň, ale budu mít spožďeni
        </input>
        <control>[APP]=Message;[CONTACT]=[Tatíčková Anna];
        [TEXT]=promiň, ale budu mít spožďeni</control>
    </sentence>
    <sentence id="23">
        <input>Napiš zprávu Monice nezapomeň koupit zmrzlinu,
        bez ní neusnu</input>
        <control>[APP]=Message;[CONTACT]=[Grofová Monika];
        [TEXT]=nezapomeň koupit zmrzlinu, bez ní neusnu</control>
```

```
    </sentence>
    <sentence id="24">
        <input>Pošli zprávu kup meloun a vodku pro Tomáše</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš, Barták Tomáš,
         ...];
        [TEXT]=kup meloun a vodku</control>
    </sentence>
    <sentence id="25">
        <input>Pošli zprávu kup meloun a vodku na číslo 775123456</input>
        <control>[APP]=Message;[PHONENUMBER]=775123456;
        [TEXT]=kup meloun a vodku</control>
    </sentence>
    <sentence id="26">
        <input>Pošli zprávu kup meloun a vodku na číslo +420775123456
        </input>
        <control>[APP]=Message;[PHONENUMBER]=+420775123456;
        [TEXT]=kup meloun a vodku</control>
    </sentence>
    <sentence id="27">
        <input>Pošli zprávu kup meloun a vodku na číslo 00420775123456
        </input>
        <control>[APP]=Message;[PHONENUMBER]=00420775123456;
        [TEXT]=kup meloun a vodku</control>
    </sentence>
    <sentence id="28">
        <input>Zavolej na číslo 775123456</input>
        <control>[APP]=Call;[PHONENUMBER]=775123456</control>
    </sentence>
    <sentence id="29">
        <input>Vytoč na číslo 00420775123456</input>
        <control>[APP]=Call;[PHONENUMBER]=00420775123456</control>
    </sentence>
    <sentence id="30">
        <input>Vytoč na číslo 775 123 456</input>
        <control>[APP]=Call;[PHONENUMBER]=775123456</control>
    </sentence>
    <sentence id="31">
        <input>Vytoč na číslo 00420 775 123 456</input>
        <control>[APP]=Call;[PHONENUMBER]=00420775123456</control>
    </sentence>
    <sentence id="31">
        <input>napiš zprávu na číslo 775 920 696 s textem Ahoj jak se
        máš</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Ahoj jak se máš</control>
    </sentence>
    <sentence id="32">
        <input>napiš zprávu Ahoj jak se máš pro číslo 775 920 696</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Ahoj jak se máš</control>
    </sentence>
    <sentence id="33">
        <input>napiš zprávu Ahoj jak se máš pro 775 920 696</input>
```

```
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Ahoj jak se máš</control>
    </sentence>
    <sentence id="34">
        <input>Napiš zprávu Ahoj jak se máš pro číslo 775 920 696</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Ahoj jak se máš</control>
    </sentence>
    <sentence id="40">
        <input>Napiš mail s textem Posílám fotky z dovolené pro Moniku
        Grofovou</input>
        <control>[APP]=Email;[CONTACT]=[Grofová Monika];
        [TEXT]=posílám fotky z dovolené</control>
    </sentence>
    <sentence id="41">
        <input>Napiš mail pro Tomáše v příloze posílám účet</input>
        <control>[APP]=Email;[CONTACT]=[Barták Tomáš, ...];
        [TEXT]=v příloze posílám účet</control>
    </sentence>
    <sentence id="42">
        <input>Email s obsahem posílám fotky z dovolené pro Radku</input>
        <control>[APP]=Email;[CONTACT]=[Grofová Radka];
        [TEXT]=posílám fotky z dovolené</control>
    </sentence>
    <sentence id="43">
        <input>Email s obsahem posílám fotky z dovolené pro Mamku</input>
        <control>[APP]=Email;[CONTACT]=[Mamka];
        [TEXT]=posílám fotky z dovolené</control>
    </sentence>
    <sentence id="50">
        <input>Jaké bude zítra počasí</input>
        <control>[APP]=Weather;[DATE]=tomorow</control>
    </sentence>
    <sentence id="51">
        <input>Počasí dneska</input>
        <control>[APP]=Weather;[DATE]=today</control>
    </sentence>
    <sentence id="52">
        <input>Počasí</input>
        <control>[APP]=Weather</control>
    </sentence>
    <sentence id="53">
        <input>Jak bude zítra</input>
        <control>[APP]=Weather;[DATE]=tomorow</control>
    </sentence>
    <sentence id="54">
        <input>Jaké počasí bude v pondělí</input>
        <control>[APP]=Weather;[DATE]=monday</control>
    </sentence>
    <sentence id="55">
        <input>Jaké počasí bude v úterý</input>
        <control>[APP]=Weather;[DATE]=tuesday</control>
    </sentence>
```

```
<sentence id="56">
    <input>Jaké počasí bude ve středu</input>
    <control>[APP]=Weather;[DATE]=wendsday</control>
</sentence>
<sentence id="57">
    <input>Jaké počasí bude v čtvrtek</input>
    <control>[APP]=Weather;[DATE]=thursday</control>
</sentence>
<sentence id="58">
    <input>Jaké bude v pátek počasí</input>
    <control>[APP]=Weather;[DATE]=friday</control>
</sentence>
<sentence id="59">
    <input>Jaké počasí bude v sobotu</input>
    <control>[APP]=Weather;[DATE]=saturday</control>
</sentence>
<sentence id="60">
    <input>Jaké počasí bude v neděli</input>
    <control>[APP]=Weather;[DATE]=sunday</control>
</sentence>
<sentence id="61">
    <input>Nastav budík na 8:00</input>
    <control>[APP]=Alarm;[TIME]=8:00</control>
</sentence>
<sentence id="62">
    <input>Nastav budík na 16:00</input>
    <control>[APP]=Alarm;[TIME]=16:00</control>
</sentence>
<sentence id="63">
    <input>Vzbuď mě v 8:00</input>
    <control>[APP]=Alarm;[TIME]=8:00</control>
</sentence>
<sentence id="64">
    <input>Probuď mě v 8:00</input>
    <control>[APP]=Alarm;[TIME]=8:00</control>
</sentence>
<sentence id="65">
    <input>Probuď mě v 23:00</input>
    <control>[APP]=Alarm;[TIME]=23:00</control>
</sentence>
<sentence id="66">
    <input>Probuď mě v 8:00 dopoledne</input>
    <control>[APP]=Alarm;[TIME]=8:00</control>
</sentence>
<sentence id="67">
    <input>Probuď mě v 8:00 odpoledne</input>
    <control>[APP]=Alarm;[TIME]=20:00</control>
</sentence>
<sentence id="68">
    <input>Probuď mě za 3 hodiny</input>
    <control>[APP]=Alarm;[TIME]=+3:00</control>
</sentence>
<sentence id="69">
```

```
        <input>Probuď mě za dvě hodiny</input>
        <control>[APP]=Alarm;[TIME]=+2:00</control>
    </sentence>
    <sentence id="70">
        <input>Nastav odpočet na jednu minutu</input>
        <control>[APP]=Timer;[TIME]=1m</control>
    </sentence>
    <sentence id="71">
        <input>Nastav odpočet na dvě minuty</input>
        <control>[APP]=Timer;[TIME]=2m</control>
    </sentence>
    <sentence id="72">
        <input>Nastav odpočet na tři minuty</input>
        <control>[APP]=Timer;[TIME]=3m</control>
    </sentence>
    <sentence id="73">
        <input>Nastav odpočet na 3 minuty</input>
        <control>[APP]=Timer;[TIME]=3m</control>
    </sentence>
    <sentence id="74">
        <input>Nastav odpočet na 4 minuty</input>
        <control>[APP]=Timer;[TIME]=4m</control>
    </sentence>
    <sentence id="75">
        <input>Nastav odpočet na 5 minut</input>
        <control>[APP]=Timer;[TIME]=5m</control>
    </sentence>
    <sentence id="80">
        <input>Spočítej 3 + 2 * 4 + 1 / 3</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="81">
        <input>Spočítej 3 plus 2 krát 4 plus 1 děleno 3</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="82">
        <input>Spočítej tři plus dva krát čtyři plus jedna děleno třemi
        </input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="83">
        <input>Tři plus dva krát čtyři plus jedna děleno třemi spočítej
        </input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="84">
        <input>Tři plus dva krát čtyři plus jedna děleno čtyřmi spočítej
        </input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/4</control>
    </sentence>
    <sentence id="85">
        <input>Kolik je tři plus dva krát čtyři plus jedna děleno čtyřmi
        </input>
```

```
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/4</control>
    </sentence>
    <sentence id="86">
        <input>Spočítej 0,5 plus 0,7</input>
        <control>[APP]=Calc;[EXAMPLE]=0,5+0,7</control>
    </sentence>
    <sentence id="90">
        <input>Nastav upozornění na čtvrtek 8 hodin</input>
        <control>[APP]=Calendar;[DATE]=Thursday;[TIME]=8:00</control>
    </sentence>
    <sentence id="91">
        <input>Nastav upozornění na čtvrtek 8 hodin odpoledne</input>
        <control>[APP]=Calendar;[DATE]=Thursday;[TIME]=20:00</control>
    </sentence>
    <sentence id="92">
        <input>Vytvoř úkol na sobotu 8 hodin odpoledne</input>
        <control>[APP]=Calendar;[DATE]=saturday;[TIME]=20:00</control>
    </sentence>
    <sentence id="93">
        <input>Přidej do kalendáře schůzku na úterý 3 hodiny odpoledne
        </input>
        <control>[APP]=Calendar;[DATE]=tuesday;[TIME]=15:00</control>
    </sentence>
    <sentence id="94">
        <input>Přidej do kalendáře schůzku na úterý 3 hodiny odpoledne
        </input>
        <control>[APP]=Calendar;[DATE]=tuesday;[TIME]=15:00</control>
    </sentence>
    <sentence id="95">
        <input>Přidej do kalendáře schůzku na 12.4.2017 11 hodiny
        dopoledne</input>
        <control>[APP]=Calendar;[DATE]=2017-4-12;[TIME]=11:00</control>
    </sentence>
    <sentence id="96">
        <input>Přidej do kalendáře schůzku na 18.4. v 9 hodiny
        dopoledne</input>
        <control>[APP]=Calendar;[DATE]=2017-4-18;[TIME]=9:00</control>
    </sentence>
    <sentence id="97">
        <input>Přidej do kalendáře schůzku na 31.5 v 10 hodiny
        dopoledne</input>
        <control>[APP]=Calendar;[DATE]=2017-5-31;[TIME]=10:00</control>
    </sentence>
    <sentence id="98">
        <input>Přidej do kalendáře schůzku na 31. dubna v 10 hodiny
        dopoledne</input>
        <control>[APP]=Calendar;[DATE]=2017-4-31;[TIME]=10:00</control>
    </sentence>
    <sentence id="99">
        <input>Přidej do kalendáře schůzku na 31 dubna 2018 v 10 hodiny
        dopoledne</input>
        <control>[APP]=Calendar;[DATE]=2018-4-31;[TIME]=10:00</control>
    </sentence>
```

```
<sentence id="100">
    <input>Řekni mi vtip</input>
    <control>[APP]=Joke</control>
</sentence>
<sentence id="101">
    <input>Pověz mi vtip</input>
    <control>[APP]=Joke</control>
</sentence>
<sentence id="102">
    <input>Řekni něco srandovního</input>
    <control>[APP]=Joke</control>
</sentence>
<sentence id="103">
    <input>Vyprávěj mi nějaký příbeh</input>
    <control>[APP]=Story</control>
</sentence>
<sentence id="104">
    <input>Vyprávěj mi nějakou povídku</input>
    <control>[APP]=Story</control>
</sentence>
<sentence id="105">
    <input>Řekni mi příbeh</input>
    <control>[APP]=Story</control>
</sentence>
<sentence id="106">
    <input>Vyprávěj mi nějakou pohádku</input>
    <control>[APP]=Fairy tales</control>
</sentence>
<sentence id="107">
    <input>Řekni mi pohádku</input>
    <control>[APP]=Fairy tales</control>
</sentence>
<sentence id="110">
    <input>Otevři aplikaci volání</input>
    <control>[APP]=Call</control>
</sentence>
<sentence id="111">
    <input>Otevři aplikaci zprávy</input>
    <control>[APP]=Message</control>
</sentence>
<sentence id="112">
    <input>Otevři aplikaci kontakty</input>
    <control>[APP]=Contacts</control>
</sentence>
<sentence id="113">
    <input>Otevři nastavení</input>
    <control>[APP]=Setting</control>
</sentence>
<sentence id="114">
    <input>Otevři informace o telefonu</input>
    <control>[APP]=Info</control>
</sentence>
<sentence id="115">
```

```
        <input>Otevři oblíbené</input>
        <control>[APP]=Favourites</control>
    </sentence>
    <sentence id="116">
        <input>Otevři hudební přehrávač</input>
        <control>[APP]=Player</control>
    </sentence>
    <sentence id="117">
        <input>Otevři rádio</input>
        <control>[APP]=Radio</control>
    </sentence>
    <sentence id="118">
        <input>Otevři knihy</input>
        <control>[APP]=Books</control>
    </sentence>
    <sentence id="119">
        <input>Otevři android aplikace</input>
        <control>[APP]=AndroidApp</control>
    </sentence>
    <sentence id="120">
        <input>Otevři rozpoznávač barev</input>
        <control>[APP]=ColorsRekognizer</control>
    </sentence>
    <sentence id="121">
        <input>Otevři lupu</input>
        <control>[APP]=MagnifyingGlass</control>
    </sentence>
    <sentence id="122">
        <input>Otevři poznámky</input>
        <control>[APP]=Notes</control>
    </sentence>
    <sentence id="123">
        <input>Otevři záznamník</input>
        <control>[APP]=Recorder</control>
    </sentence>
    <sentence id="124">
        <input>Otevři wifi</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=OPEN</control>
    </sentence>
    <sentence id="125">
        <input>Otevři rozpoznávání bankovek</input>
        <control>[APP]=BankRecognizer</control>
    </sentence>
    <sentence id="126">
        <input>Vypni telefony</input>
        <control>[APP]=ShutDown</control>
    </sentence>
    <sentence id="127">
        <input>Otevři manuál</input>
        <control>[APP]=Manual</control>
    </sentence>
    <sentence id="128">
        <input>Budík</input>
```

```
        <control>[APP]=Alarm</control>
    </sentence>
    <sentence id="129">
        <input>Nastav budík na 8 hodin 45 minut</input>
        <control>[APP]=Alarm;[TIME]=8:45</control>
    </sentence>
    <sentence id="130">
        <input>Napiš poznámku ve tvaru Nezapomenout koupit mlíko</input>
        <control>[APP]=Notes;[TEXT]=Nezapomenout koupit mlíko</control>
    </sentence>
    <sentence id="131">
        <input>Napiš poznámku s textem Nezapomenout koupit mlíko</input>
        <control>[APP]=Notes;[TEXT]=Nezapomenout koupit mlíko</control>
    </sentence>
    <sentence id="132">
        <input>Napiš poznámku Nezapomenout koupit mlíko</input>
        <control>[APP]=Notes;[TEXT]=Nezapomenout koupit mlíko</control>
    </sentence>
    <sentence id="140">
        <input>Spusť rádio Beat</input>
        <control>[APP]=Radio;[STATION]=[Beat]</control>
    </sentence>
    <sentence id="141">
        <input>Zapni rádio Hey</input>
        <control>[APP]=Radio;[STATION]=[Hey]</control>
    </sentence>
    <sentence id="142">
        <input>Přehraj rádio Evropa 2</input>
        <control>[APP]=Radio;[STATION]=[Evropa 2]</control>
    </sentence>
    <sentence id="150">
        <input>Přehraj písničku Crossroads</input>
        <control>[APP]=Player;[ARTIST]=The Offspring;
        [ALBUM]=The Offspring;
        [GENRE]=Alternative rock;[NAME]=Crossroads</control>
    </sentence>
    <sentence id="151">
        <input>Přehraj něco od Offspring</input>
        <control>[APP]=Player;[ARTIST]=The Offspring;
        [ALBUM]=The Offspring;
        </control>
    </sentence>
    <sentence id="152">
        <input>Pusť nějaký pop</input>
        <control>[APP]=Player;[GENRE]=Pop</control>
    </sentence>
    <sentence id="153">
        <input>Přehraj písničku Love Yourself</input>
        <control>[APP]=Player;[ARTIST]=Justin Bieber;[ALBUM]=Purpose;
        [GENRE]=Pop;[NAME]=Love Yourself</control>
    </sentence>
    <sentence id="154">
        <input>Přehraj album Songs of Innocence</input>
```

```
        <control>[APP]=Player;[ARTIST]=U2;[ALBUM]=Songs of Innocence
        </control>
    </sentence>
    <sentence id="160">
        <input>Stav baterie</input>
        <control>[APP]=Phone;[STATUS]=BATTERY</control>
    </sentence>
    <sentence id="161">
        <input>Stav signálu</input>
        <control>[APP]=Phone;[STATUS]=SIGNAL</control>
    </sentence>
    <sentence id="162">
        <input>Baterie</input>
        <control>[APP]=Phone;[STATUS]=BATTERY</control>
    </sentence>
    <sentence id="163">
        <input>Signál</input>
        <control>[APP]=Phone;[STATUS]=SIGNAL</control>
    </sentence>
    <sentence id="164">
        <input>Stav wifi</input>
        <control>[APP]=Phone;[STATUS]=WIFI</control>
    </sentence>
    <sentence id="165">
        <input>wifi vypnout</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=OFF</control>
    </sentence>
    <sentence id="166">
        <input>Zapnout wifi</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=ON;</control>
    </sentence>
    <sentence id="167">
        <input>wifi zapnout</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=ON;</control>
    </sentence>
    <sentence id="170">
        <input>Kolik je hodin?</input>
        <control>[APP]=Info;[CATEGORY]=TIME</control>
    </sentence>
    <sentence id="171">
        <input>Kolik je?</input>
        <control>[APP]=Info;[CATEGORY]=TIME</control>
    </sentence>
    <sentence id="173">
        <input>Co je dnes za den?</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="174">
        <input>Co je za den?</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="175">
        <input>Restartovat telefon</input>
```

```
        <control>[APP]=Restart</control>
    </sentence>
    <sentence id="176">
        <input>Restartuj telefon</input>
        <control>[APP]=Restart</control>
    </sentence>
    <sentence id="180">
        <input>Otevři aplikaci facebook</input>
        <control>[APP]=Facebook</control>
    </sentence>
    <sentence id="181">
        <input>Otevři aplikaci TapTapSee</input>
        <control>[APP]=TapTapSee</control>
    </sentence>
    <sentence id="182">
        <input>Otevři aplikaci Facebook</input>
        <control>[APP]=Facebook</control>
    </sentence>
    <sentence id="183">
        <input>Otevři aplikaci WhatsApp</input>
        <control>[APP]=WhatsApp</control>
    </sentence>
    <sentence id="184">
        <input>Otevři internet</input>
        <control>[APP]=InternetBrowser</control>
    </sentence>
    <sentence id="185">
        <input>Otevři internetový prohlížeč</input>
        <control>[APP]=InternetBrowser</control>
    </sentence>
    <sentence id="186">
        <input>Najdi na internetu kdy zemřel Karel IV</input>
        <control>[APP]=InternetBrowser;[TEXT]=kdy zemřel Karel IV
        </control>
    </sentence>
    <sentence id="187">
        <input>Vyhledej Jak je vysoká Eifelova věž</input>
        <control>[APP]=InternetBrowser;[TEXT]=jak je vysoká Eifelova věž
        </control>
    </sentence>
    <sentence id="188">
        <input>Najdi Jak hrál včera manchester united</input>
        <control>[APP]=InternetBrowser;[TEXT]=jak hrál včera manchester
        united</control>
    </sentence>
    <sentence id="189">
        <input>Najdi na internetu Kdy bude hrát Manchester United</input>
        <control>[APP]=InternetBrowser;[TEXT]=Kdy bude hrát Manchester
        United</control>
    </sentence>
    <sentence id="190">
        <input>Zruš budík na 8:00</input>
        <control>[APP]=Alarm;[TIME]=8:00;[ACTION]=DESTROY</control>
```

```
    </sentence>
    <sentence id="191">
        <input>Jaké bude počasí pozítří</input>
        <control>[APP]=Weather;[DATE]=afterTomorow</control>
    </sentence>
    <sentence id="192">
        <input>Otevři jízdní řády</input>
        <control>[APP]=Timetable</control>
    </sentence>
    <sentence id="193">
        <input>Kolikátého dnes je</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="200">
        <input>Vytvoř úkol na sobotu 8 hodin odpoledne s názvem
        Fotbal</input>
        <control>[APP]=Calendar;[TEXT]=Fotbal;[DATE]=saturday;
        [TIME]=20:00</control>
    </sentence>
    <sentence id="201">
        <input>Vytvoř úkol s názvem Fotbal na sobotu 8 hodin
        odpoledne</input>
        <control>[APP]=Calendar;[TEXT]=Fotbal;[DATE]=saturday;
        [TIME]=20:00</control>
    </sentence>
    <sentence id="202">
        <input>Vytvoř úkol Dentalní higiena na sobotu 8 hodin
        odpoledne</input>
        <control>[APP]=Calendar;[TEXT]=Dentalní higiena;[DATE]=saturday;
        [TIME]=20:00</control>
    </sentence>
    <sentence id="203">
        <input>Napiš zprávu s textem Kup kopu vajec a chilli pro kontakt
        Tomáš Barták</input>
        <control>[APP]=Message;[CONTACT]=[Barták Tomáš];
        [TEXT]=Kup kopu vajec a chilli</control>
    </sentence>
    <sentence id="204">
        <input>Zavolej Tomáše Blažka na mobil</input>
        <control>[APP]=Call;[CONTACT]=[Blažek Tomáš]</control>
    </sentence>
    <sentence id="205">
        <input>Zavolej Grofovou do práce</input>
        <control>[APP]=Call;[CONTACT]=[Grofová Monča, Grofová Radka, ...]
        </control>
    </sentence>
    <sentence id="206">
        <input>Vytoč Tomáše domů</input>
        <control>[APP]=Call;[CONTACT]=[Blažek Tomáš, Barták Tomáš, ...]
        </control>
    </sentence>
    <sentence id="207">
        <input>Zavolej Blažkovi do práce</input>
```

```
        <control>[APP]=Call;[CONTACT]=[Blažek Tomáš]</control>
    </sentence>
    <sentence id="208">
        <input>Pošli Tomášovi Blažkovi na mobil zprávu že má koupit
        3 vejce, mléko,
        špagety a nějaké pečivo</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš];
        [TEXT]=že má koupit 3 vejce,
        mléko, špagety a nějaké pečivo</control>
    </sentence>
    <sentence id="210">
        <input>Napiš Aničce Tatíčkové na mobil promiň, ale budu mít
        spožděni </input>
        <control>[APP]=Message;[CONTACT]=[Tatíčková Anna];
        [TEXT]=promiň,
        ale budu mít spožděni</control>
    </sentence>
    <sentence id="211">
        <input>Napiš zprávu Monice domů nezapomeň koupit zmrzlinu,
        bez ní neusnu</input>
        <control>[APP]=Message;[CONTACT]=[Grofová Monika];
        [TEXT]=nezapomeň koupit zmrzlinu,
        bez ní neusnu</control>
    </sentence>
    <sentence id="212">
        <input>Pošli zprávu kup meloun a vodku pro Tomáše na
        mobil</input>
        <control>[APP]=Message;[CONTACT]=[Blažek Tomáš,
        Barták Tomáš, ...];
        [TEXT]=kup meloun a vodku</control>
    </sentence>
    <sentence id="213">
        <input>Napiš Petr Kohout zprávu na mobil jestli to
        funguje tak na to s***</input>
        <control>[APP]=Message;[CONTACT]=[Kohout Petr;
        [TEXT]=jestli to
        funguje tak na to s***</control>
    </sentence>
</resources>
```

H.2 Anglická trénovací data

```
<resources>
    <sentence id="0">
        <input>Send message for Tomas Green</input>
        <control>[APP]=Message;[CONTACT]=[Green Tomas];
        [TEXT]=null</control>
    </sentence>
    <sentence id="1">
        <input>Call Tomas Muller</input>
        <control>[APP]=Call;[CONTACT]=[Muller Tomas]</control>
    </sentence>
```

```
<sentence id="2">
    <input>Call Tomas</input>
    <control>[APP]=Call;[CONTACT]=[Green Thomas, Muller Tomas, ...]
    </control>
</sentence>
<sentence id="3">
    <input>Dial Tomas</input>
    <control>[APP]=Call;[CONTACT]=[Green Thomas, Muller Tomas, ...]
    </control>
</sentence>
<sentence id="4">
    <input>Call McDonnald</input>
    <control>[APP]=Call;[CONTACT]=[McDonnald Tomas]</control>
</sentence>
<sentence id="5">
    <input>Write message for Tomas Brown with text i'm really
    looking forward to the holidays</input>
    <control>[APP]=Message;[CONTACT]=[Brown Tomas];
    [TEXT]=i'm really looking forward to the holidays</control>
</sentence>
<sentence id="6">
    <input>Write message for George Cloney where text is i'm really
    looking forward to the holidays</input>
    <control>[APP]=Message;[CONTACT]=[Cloney George];
    [TEXT]=i'm really looking forward to the holidays</control>
</sentence>
<sentence id="7">
    <input>Write message with text Buy kick eggs and chilli
    for Jana Marry</input>
    <control>[APP]=Message;[CONTACT]=[Marry Jane];
    [TEXT]=Buy kick eggs and chilli</control>
</sentence>
<sentence id="8">
    <input>Write message Buy kick eggs and chilli for Marry Jana
    </input>
    <control>[APP]=Message;[CONTACT]=[Marry Jane];
    [TEXT]=Buy kick eggs and chilli</control>
</sentence>
<sentence id="9">
    <input>Write sms Buy kick eggs and chilli for Anna Jones
    </input>
    <control>[APP]=Message;[CONTACT]=[Jones Anna];
    [TEXT]=Buy kick eggs and chilli</control>
</sentence>
<sentence id="10">
    <input>Write sms George Cloney Buy kick eggs and chilli
    </input>
    <control>[APP]=Message;[CONTACT]=[Cloney George];
    [TEXT]=Buy kick eggs and chilli</control>
</sentence>
<sentence id="11">
    <input>Write message for Tomas John Buy kick eggs and chilli
    for my dad</input>
```

```
        <control>[APP]=Message;[CONTACT]=[John Tomas];
        [TEXT]=Buy kick eggs and chilli for my dad</control>
    </sentence>
    <sentence id="12">
        <input>Write message for Tomas John with text Buy kick eggs
        and chilli for my dad</input>
        <control>[APP]=Message;[CONTACT]=[John Tomas];
        [TEXT]=Buy kick eggs and chilli for my dad</control>
    </sentence>
    <sentence id="13">
        <input>Write message with text Buy kick eggs and chilli for
        my dad for Tomas John</input>
        <control>[APP]=Message;[CONTACT]=[John Tomas];
        [TEXT]=Buy kick eggs and chilli for my dad</control>
    </sentence>
    <sentence id="14">
        <input>Write George text I am in a tram,
        I'll be there in 15 minutes</input>
        <control>[APP]=Message;[CONTACT]=[Cloney George, ...];
        [TEXT]=I am in a tram, I'll be there in 15 minutes</control>
    </sentence>
    <sentence id="15">
        <input>Send message to Tomas</input>
        <control>[APP]=Message;[CONTACT]=[John Toma, Brown Tomas, ...];
        [TEXT]=null</control>
    </sentence>
    <sentence id="16">
        <input>Send message to Marry Jane with Greetings from Italy
        we will have nice sun is shining and it's warm</input>
        <control>[APP]=Message;[CONTACT]=[Marry Jane];
        [TEXT]=Greetings from Italy we will have nice sun
        is shining and it's warm</control>
    </sentence>
    <sentence id="17">
        <input>Send sms Tomas Brown where text is Buy some fish and
        frozen vegetables I'll make a pizza</input>
        <control>[APP]=Message;[CONTACT]=[Brown Tomas];
        [TEXT]=Buy some fish and frozen vegetables I'll make a pizza
        </control>
    </sentence>
    <sentence id="20">
        <input>Send message for Kate Hugse Buy 3 eggs, milk,
        spaghetti and some pastries</input>
        <control>[APP]=Message;[CONTACT]=[Hugse Kate];
        [TEXT]=Buy 3 eggs, milk, spaghetti and some pastries
        </control>
    </sentence>
    <sentence id="21">
        <input>Send message Beer I'm late to
        Kate Midleton</input>
        <control>[APP]=Message;[CONTACT]=[Midleton Kate];
        [TEXT]=Beer I'm late</control>
    </sentence>
```

```
<sentence id="22">
    <input>Send Lucy Lein I'm sorry but I have delayed </input>
    <control>[APP]=Message;[CONTACT]=[Lein Lucy];
    [TEXT]=I'm sorry but I have delayed</control>
</sentence>
<sentence id="23">
    <input>Send message Kate Do not forget to buy ice cream,
    without going to sleep</input>
    <control>[APP]=Message;[CONTACT]=[Midleton Kate, ...];
    [TEXT]=Do not forget to buy ice cream, without going to sleep
    </control>
</sentence>
<sentence id="24">
    <input>Write message to Bush George Buy watermelon
    and vodka for Tomas</input>
    <control>[APP]=Message;[CONTACT]=[Bush George];
    [TEXT]=Buy watermelon and vodka for Tomas</control>
</sentence>
<sentence id="25">
    <input>Write message Buy watermelon and vodka
    for number 775123456</input>
    <control>[APP]=Message;[PHONENUMBER]=775123456;
    [TEXT]=Buy watermelon and vodka</control>
</sentence>
<sentence id="26">
    <input>Write message Buy watermelon and vodka
    for number +420775123456</input>
    <control>[APP]=Message;[PHONENUMBER]=+420775123456;
    [TEXT]=Buy watermelon and vodka</control>
</sentence>
<sentence id="27">
    <input>Write message Buy watermelon and vodka
    for number 00420775123456</input>
    <control>[APP]=Message;[PHONENUMBER]=00420775123456;
    [TEXT]=Buy watermelon and vodka</control>
</sentence>
<sentence id="28">
    <input>Call to number 775123456</input>
    <control>[APP]=Call;[PHONENUMBER]=775123456</control>
</sentence>
<sentence id="29">
    <input>Call to number 00420775123456</input>
    <control>[APP]=Call;[PHONENUMBER]=00420775123456</control>
</sentence>
<sentence id="30">
    <input>Call to number 775 123 456</input>
    <control>[APP]=Call;[PHONENUMBER]=775123456</control>
</sentence>
<sentence id="31">
    <input>Call to number 00420 775 123 456</input>
    <control>[APP]=Call;
    [PHONENUMBER]=00420775123456</control>
</sentence>
```

```
<sentence id="40">
    <input>Write mail with text sending vacation photos for mother
    </input>
    <control>[APP]=Email;[CONTACT]=[Mother];
    [TEXT]=sending vacation photos</control>
</sentence>
<sentence id="41">
    <input>Send mail for Tomas in annex sending account</input>
    <control>[APP]=Email;[CONTACT]=[Brown Tomas, ...];
    [TEXT]=in annex sending account</control>
</sentence>
<sentence id="42">
    <input>Email containing send vacation photos for Rita</input>
    <control>[APP]=Email;[CONTACT]=[Steel Rita];
    [TEXT]=send vacation photos</control>
</sentence>
<sentence id="50">
    <input>What the weather will be tomorrow</input>
    <control>[APP]=Weather;[DATE]=tomorow</control>
</sentence>
<sentence id="51">
    <input>What the weather will be today</input>
    <control>[APP]=Weather;[DATE]=today</control>
</sentence>
<sentence id="54">
    <input>What the weather will be monday</input>
    <control>[APP]=Weather;[DATE]=monday</control>
</sentence>
<sentence id="55">
    <input>What the weather will be tuesday</input>
    <control>[APP]=Weather;[DATE]=tuesday</control>
</sentence>
<sentence id="56">
    <input>What the weather will be wendsday</input>
    <control>[APP]=Weather;[DATE]=wendsday</control>
</sentence>
<sentence id="57">
    <input>What the weather will be thursday</input>
    <control>[APP]=Weather;[DATE]=thursday</control>
</sentence>
<sentence id="58">
    <input>What the weather will be friday</input>
    <control>[APP]=Weather;[DATE]=friday</control>
</sentence>
<sentence id="59">
    <input>What the weather will be saturday</input>
    <control>[APP]=Weather;[DATE]=saturday</control>
</sentence>
<sentence id="60">
    <input>What the weather will be sunday</input>
    <control>[APP]=Weather;[DATE]=sunday</control>
</sentence>
<sentence id="61">
```

```
        <input>Set the alarm for 8:00</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="62">
        <input>Set the alarm clock for 16:00</input>
        <control>[APP]=Alarm;[TIME]=16:00</control>
    </sentence>
    <sentence id="63">
        <input>Wake me up at 8:00</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="64">
        <input>Wake me up at 8:00 am</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="65">
        <input>Set the alarm for 11:00 PM</input>
        <control>[APP]=Alarm;[TIME]=23:00</control>
    </sentence>
    <sentence id="66">
        <input>Wake me up at 8:00 AM</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="67">
        <input>Set the alarm for 8:00 pm</input>
        <control>[APP]=Alarm;[TIME]=20:00</control>
    </sentence>
    <sentence id="68">
        <input>Wake me up after 3 hours</input>
        <control>[APP]=Alarm;[TIME]=+3h</control>
    </sentence>
    <sentence id="69">
        <input>Wake me up in 2 hours</input>
        <control>[APP]=Alarm;[TIME]=+2h</control>
    </sentence>
    <sentence id="70">
        <input>Set countdown for one minute</input>
        <control>[APP]=Timer;[TIME]=1m</control>
    </sentence>
    <sentence id="71">
        <input>Set your timer for two minute</input>
        <control>[APP]=Timer;[TIME]=2m</control>
    </sentence>
    <sentence id="72">
        <input>Set your timer for 3 minute</input>
        <control>[APP]=Timer;[TIME]=3m</control>
    </sentence>
    <sentence id="73">
        <input>Set your countdown na three minute</input>
        <control>[APP]=Timer;[TIME]=3m</control>
    </sentence>
    <sentence id="74">
        <input>Set timer on 4 minute</input>
```

```
        <control>[APP]=Timer;[TIME]=4m</control>
    </sentence>
    <sentence id="75">
        <input>Set your countdown for 5 minute</input>
        <control>[APP]=Timer;[TIME]=5m</control>
    </sentence>
    <sentence id="80">
        <input>Compute 3 + 2 * 4 + 1 / 3</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="81">
        <input>Compute 3 plus 2 times 4 plus 1 divide 3</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="82">
        <input>Compute three plus two times four plus one divided three
        </input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="90">
        <input>Make a task to Sunday 6 pm</input>
        <control>[APP]=Calendar;[DATE]=sunday;[TIME]=18:00</control>
    </sentence>
    <sentence id="91">
        <input>Create a task to Thursday 8 am</input>
        <control>[APP]=Calendar;[DATE]=thursday;[TIME]=8:00</control>
    </sentence>
    <sentence id="92">
        <input>Create a task to Saturday 8 pm</input>
        <control>[APP]=Calendar;[DATE]=saturday;[TIME]=20:00</control>
    </sentence>
    <sentence id="93">
        <input>Add to calendar appointment on Tuesdays, 3 pm</input>
        <control>[APP]=Calendar;[DATE]=tuesday;[TIME]=15:00</control>
    </sentence>
    <sentence id="94">
        <input>Add to calendar appointment on Tuesdays, 3 am</input>
        <control>[APP]=Calendar;[DATE]=tuesday;[TIME]=3:00</control>
    </sentence>
    <sentence id="95">
        <input>Add meeting to calendar on 4.12.2017 for 11 am</input>
        <control>[APP]=Calendar;[DATE]=2017-4-12;[TIME]=11:00</control>
    </sentence>
    <sentence id="96">
        <input>Add meeting to calendar on 4.18. for 9 am</input>
        <control>[APP]=Calendar;[DATE]=2017-4-18;[TIME]=9:00</control>
    </sentence>
    <sentence id="97">
        <input>Add meeting to calendar on 5.31 at 10 AM</input>
        <control>[APP]=Calendar;[DATE]=2017-5-31;[TIME]=10:00</control>
    </sentence>
    <sentence id="98">
        <input>Add meeting to calendar on April 31 at 10 am</input>
```

```
        <control>[APP]=Calendar;[DATE]=2017-4-31;[TIME]=10:00</control>
    </sentence>
    <sentence id="101">
        <input>Tell me a joke</input>
        <control>[APP]=Joke</control>
    </sentence>
    <sentence id="102">
        <input>Sai me a joke</input>
        <control>[APP]=Joke</control>
    </sentence>
    <sentence id="103">
        <input>Tell me a some funny story</input>
        <control>[APP]=Joke</control>
    </sentence>
    <sentence id="104">
        <input>Tell me a story</input>
        <control>[APP]=Story</control>
    </sentence>
    <sentence id="106">
        <input>Tell me a fairy tale</input>
        <control>[APP]=Fairy tales</control>
    </sentence>
    <sentence id="106">
        <input>Tell me a tale</input>
        <control>[APP]=Fairy tales</control>
    </sentence>
    <sentence id="107">
        <input>Tell me a fairy</input>
        <control>[APP]=Fairy tales</control>
    </sentence>
    <sentence id="110">
        <input>Open app calls</input>
        <control>[APP]=Call</control>
    </sentence>
    <sentence id="111">
        <input>Open app messages</input>
        <control>[APP]=Message</control>
    </sentence>
    <sentence id="112">
        <input>Open app contacts</input>
        <control>[APP]=Contacts</control>
    </sentence>
    <sentence id="113">
        <input>Open app setting</input>
        <control>[APP]=Setting</control>
    </sentence>
    <sentence id="114">
        <input>Open app information</input>
        <control>[APP]=Info</control>
    </sentence>
    <sentence id="114">
        <input>Open app manual</input>
        <control>[APP]=Manual</control>
```

```
    </sentence>
    <sentence id="115">
        <input>Open app favourites</input>
        <control>[APP]=Favourites</control>
    </sentence>
    <sentence id="116">
        <input>Open app music player</input>
        <control>[APP]=Player</control>
    </sentence>
    <sentence id="117">
        <input>Open app radio</input>
        <control>[APP]=Radio</control>
    </sentence>
    <sentence id="118">
        <input>Open app books</input>
        <control>[APP]=Books</control>
    </sentence>
    <sentence id="119">
        <input>Open android app</input>
        <control>[APP]=AndroidApp</control>
    </sentence>
    <sentence id="120">
        <input>Open app colors rekognization</input>
        <control>[APP]=ColorsRekognizer</control>
    </sentence>
    <sentence id="121">
        <input>Open app magnifying glass</input>
        <control>[APP]=MagnifyingGlass</control>
    </sentence>
    <sentence id="122">
        <input>Open app notes</input>
        <control>[APP]=Notes</control>
    </sentence>
    <sentence id="123">
        <input>Open app recorder</input>
        <control>[APP]=Recorder</control>
    </sentence>
    <sentence id="124">
        <input>Open app wifi</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=OPEN</control>
    </sentence>
    <sentence id="125">
        <input>Open app bank recognizer</input>
        <control>[APP]=BankRecognizer</control>
    </sentence>
    <sentence id="126">
        <input>Shutdown phone</input>
        <control>[APP]=ShutDown</control>
    </sentence>
    <sentence id="130">
        <input>Write a note in the shape Do not forget to buy milk
        </input>
        <control>[APP]=Notes;[TEXT]=Do not forget to buy milk</control>
```

```
    </sentence>
    <sentence id="131">
        <input>Write a note in the form Do not forget to buy milk</input>
        <control>[APP]=Notes;[TEXT]=Do not forget to buy milk</control>
    </sentence>
    <sentence id="132">
        <input>Write a note Do not forget to buy milk</input>
        <control>[APP]=Notes;[TEXT]=Do not forget to buy milk</control>
    </sentence>
    <sentence id="140">
        <input>Start radio London</input>
        <control>[APP]=Radio;[STATION]=[BBC■London]</control>
    </sentence>
    <sentence id="141">
        <input>Play radio Cambridge</input>
        <control>[APP]=Radio;[STATION]=[Cambridge■105]</control>
    </sentence>
    <sentence id="142">
        <input>Radio TalkSport</input>
        <control>[APP]=Radio;[STATION]=[TalkSport]</control>
    </sentence>
    <sentence id="150">
        <input>Play song Crossroads</input>
        <control>[APP]=Player;[ARTIST]=The Offspring;
        [ALBUM]=The Offspring;[GENRE]=Alternative rock;[NAME]=Crossroads
        </control>
    </sentence>
    <sentence id="151">
        <input>Play some music from Offspring</input>
        <control>[APP]=Player;[ARTIST]=The Offspring;
        [ALBUM]=The Offspring;</control>
    </sentence>
    <sentence id="152">
        <input>Play pop music</input>
        <control>[APP]=Player;[GENRE]=Pop</control>
    </sentence>
    <sentence id="153">
        <input>Play song Love Yourself</input>
        <control>[APP]=Player;[ARTIST]=Justin Bieber;
        [ALBUM]=Purpose;[GENRE]=Pop;[NAME]=Love Yourself</control>
    </sentence>
    <sentence id="154">
        <input>Play album Songs of Innocence</input>
        <control>[APP]=Player;[ARTIST]=U2;[ALBUM]=Songs of Innocence
        </control>
    </sentence>
    <sentence id="160">
        <input>Battery status</input>
        <control>[APP]=Phone;[STATUS]=BATTERY</control>
    </sentence>
    <sentence id="161">
        <input>Signal status</input>
        <control>[APP]=Phone;[STATUS]=SIGNAL</control>
```

```
    </sentence>
    <sentence id="170">
        <input>What time is it?</input>
        <control>[APP]=Info;[CATEGORY]=TIME</control>
    </sentence>
    <sentence id="171">
        <input>What time do you have?</input>
        <control>[APP]=Info;[CATEGORY]=TIME</control>
    </sentence>
    <sentence id="173">
        <input>What is the day today?</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="174">
        <input>What day it is?</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="175">
        <input>Set alarm on 8:22 every wednesday</input>
        <control>[APP]=Alarm;[DAYINWEEK]=3;[TIME]=8:22</control>
    </sentence>
    <sentence id="176">
        <input>Set alarm on 22:11 every sunday</input>
        <control>[APP]=Alarm;[DAYINWEEK]=7;[TIME]=22:11</control>
    </sentence>
    <sentence id="177">
        <input>Set alarm on 22:11 every saturday</input>
        <control>[APP]=Alarm;[DAYINWEEK]=6;[TIME]=22:11</control>
    </sentence>
</resources>
```

H.3 Německá trénovací data

```
<resources>
    <sentence id="0">
        <input>Schreibe eine SMS an Thomas Müller</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=null</control>
    </sentence>
    <sentence id="1">
        <input>Rufe Thomas Müller an</input>
        <control>[APP]=Call;[CONTACT]=[Müller Thomas]
        </control>
    </sentence>
    <sentence id="2">
        <input>Rufe Graf an</input>
        <control>[APP]=Call;[CONTACT]=[Graf Marta,
        Graf Renate, ...]</control>
    </sentence>
    <sentence id="3">
        <input>Wähle Thomas</input>
        <control>[APP]=Call;[CONTACT]=[Müller Thomas,
```

```
        Schulze Thomas, ...]</control>
    </sentence>
    <sentence id="4">
        <input>Rufe Müller</input>
        <control>[APP]=Call;[CONTACT]=[Müller Thomas]</control>
    </sentence>
    <sentence id="5">
        <input>Schreibe eine SMS an Thomas Müller mit dem
        Text Ich freue mich rießig auf Urlaub</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=Ich freue mich rießig auf Urlaub</control>
    </sentence>
    <sentence id="6">
        <input>Schreibe eine SMS an Thomas Müller in der
        steht Ich freue mich rießig auf Urlaub</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=Ich freue mich rießig auf Urlaub</control>
    </sentence>
    <sentence id="7">
        <input>Schreibe eine SMS mit dem Text Kaufe einen
        Haufen Eier und Chilli für Thomas Schulze</input>
        <control>[APP]=Message;[CONTACT]=[Schulze Thomas];
        [TEXT]=Kaufe einen Haufen Eier und Chilli</control>
    </sentence>
    <sentence id="8">
        <input>Schreibe Mitteilung Kaufe einen Haufen Eier
        und Chilli für Thomas Schulze</input>
        <control>[APP]=Message;[CONTACT]=[Schulze Thomas];
        [TEXT]=Kaufe einen Haufen Eier und Chilli</control>
    </sentence>
    <sentence id="9">
        <input>Schreibe Mitteilung Kaufe einen Haufen Eier
        und Chilli für Schulze Thomas</input>
        <control>[APP]=Message;[CONTACT]=[Schulze Thomas];
        [TEXT]=Kaufe einen Haufen Eier und Chilli</control>
    </sentence>
    <sentence id="10">
        <input>Schreibe eine SMS Thomas Müller Kaufe einen
        Haufen Eier und Chilli</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=Kaufe einen Haufen Eier und Chilli</control>
    </sentence>
    <sentence id="11">
        <input>Schreibe eine SMS an Müller Thomas Kaufe
        einen Haufen Eier für Papa</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=Kaufe einen Haufen Eier für Papa</control>
    </sentence>
    <sentence id="12">
        <input>Schreibe eine SMS für Müller Thomas mit dem
        Text Kaufe einen Haufen Eier für Papa</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=Kaufe einen Haufen Eier für Papa</control>
```

```
    </sentence>
    <sentence id="13">
        <input>Schreibe eine SMS mit dem Text Kaufe einen
        Haufen Eier für Papa für Müller Thomas</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=Kaufe einen Haufen Eier für Papa</control>
    </sentence>
    <sentence id="14">
        <input>Schreibe Jan eine SMS bin in der Bahn, werde
        in 15 Minuten da sein</input>
        <control>[APP]=Message;[CONTACT]=[Ruchert Jan, ...];
        [TEXT]=bin in der Bahn, werde in 15 Minuten da sein</control>
    </sentence>
    <sentence id="15">
        <input>Schreibe Thomas eine SMS</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas,
        Schulze Thomas, ...];[TEXT]=null</control>
    </sentence>
    <sentence id="16">
        <input>Schreibe eine SMS Sarah Keidel in Form von Schicke Grüße
        aus Italien Es geht es uns hier prima Sonne scheint und es ist
        warm</input>
        <control>[APP]=Message;[CONTACT]=[Keidel Sarah];
        [TEXT]=Schicke Grüße aus Italien Es geht es uns hier prima
        Sonne scheint und es ist warm</control>
    </sentence>
    <sentence id="17">
        <input>Schreibe eine SMS Thomas Neumann in Form von
        Kaufe irgendein Fisch und Tiefkühlgemüse, ich mache daraus
        Pizza.</input>
        <control>[APP]=Message;[CONTACT]=[Neumann Thomas];
        [TEXT]=Kaufe irgendein Fisch und Tiefkühlgemüse,
        ich mache daraus Pizza.</control>
    </sentence>
    <sentence id="18">
        <input>Sende eine SMS Thomas Meier Hi Tom, hole noch paar
        Bierchen, ich werde Kartoffelpuffer machen, also kaufe
        auch noch Kartoffeln und Majoran. Beeile dich!</input>
        <control>[APP]=Message;[CONTACT]=[Meier Thomas];
        [TEXT]=Hi Tom, hole noch paar Bierchen, ich werde Kartoffelpuffer
        machen, also kaufe auch noch Kartoffeln und Majoran. Beeile dich!
        </control>
    </sentence>
    <sentence id="19">
        <input>Schreibe Thomas Meier Hallo, kaufe zum Abendbrot
        halbes Kilo Rinderhack, Spagetti, zwei Möhren, Sellerie,
        drei Petersilienwurzel, geschälte Tomaten, Butter und Weißwein.
        Den was ich mag. Und vergiss nicht zweihundert Gramm Schinken.
        Danke.</input>
        <control>[APP]=Message;[CONTACT]=[Meier Thomas];
        [TEXT]=Hallo, kaufe zum Abendbrot halbes Kilo Rinderhack,
        Spagetti, zwei Möhren, Sellerie, drei Petersilienwurzel,
        geschälte Tomaten, Butter und Weißwein. Den was ich mag.
```

```
        Und vergiss nicht zweihundert Gramm Schinken. Danke.
        </control>
    </sentence>
    <sentence id="20">
        <input>Sende Thomas Müller eine SMS dass er 3 Eier, Milch,
        Spagetti und irgendwelches Brot kauft</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas];
        [TEXT]=dass er 3 Eier, Milch, Spagetti und irgendwelches
        Brot kauft</control>
    </sentence>
    <sentence id="21">
        <input>Schreibe Jan Ruchert SMS auf Bierchen komme ich zu spät
        </input>
        <control>[APP]=Message;[CONTACT]=[Ruchert Jan];
        [TEXT]=auf Bierchen komme ich zu spät</control>
    </sentence>
    <sentence id="22">
        <input>Schreibe Anna Thümmel entschuldige, ich werde
        Verspätung haben</input>
        <control>[APP]=Message;[CONTACT]=[Thümmel Anne];
        [TEXT]=entschuldige, ich werde Verspätung haben</control>
    </sentence>
    <sentence id="23">
        <input>Schreibe eine Nachricht Marta vergiss nicht
        Eis zu kaufen, ohne schlafe ich nicht ein</input>
        <control>[APP]=Message;[CONTACT]=[Graf Marta];
        [TEXT]=vergiss nicht Eis zu kaufen, ohne schlafe ich nicht ein
        </control>
    </sentence>
    <sentence id="24">
        <input>Sende Nachricht kaufe Melone und Vodka an Thomas</input>
        <control>[APP]=Message;[CONTACT]=[Müller Thomas,
        Schulze Thomas, ...];[TEXT]=kaufe Melone und Vodka</control>
    </sentence>
    <sentence id="25">
        <input>Sende Nachricht kaufe Melone und Vodka an
        Nummer 775123456</input>
        <control>[APP]=Message;[PHONENUMBER]=775123456;
        [TEXT]=kaufe Melone und Vodka</control>
    </sentence>
    <sentence id="26">
        <input>Sende Nachricht kaufe Melone und Vodka an
        Nummer +420775123456</input>
        <control>[APP]=Message;[PHONENUMBER]=+420775123456;
        [TEXT]=kaufe Melone und Vodka</control>
    </sentence>
    <sentence id="27">
        <input>Sende Nachricht kaufe Melone und Vodka an
        Nummer 00420775123456</input>
        <control>[APP]=Message;[PHONENUMBER]=00420775123456;
        [TEXT]=kaufe Melone und Vodka</control>
    </sentence>
    <sentence id="28">
```

```
        <input>Rufe Nummer 775123456 an</input>
        <control>[APP]=Call;[PHONENUMBER]=775123456</control>
    </sentence>
    <sentence id="29">
        <input>Wähle Nummer 00420775123456</input>
        <control>[APP]=Call;[PHONENUMBER]=00420775123456</control>
    </sentence>
    <sentence id="30">
        <input>Wähle Nummer 775 123 456</input>
        <control>[APP]=Call;[PHONENUMBER]=775123456
        </control>
    </sentence>
    <sentence id="31">
        <input>Wähle Nummer 00420 775 123 456</input>
        <control>[APP]=Call;[PHONENUMBER]=00420775123456
        </control>
    </sentence>
    <sentence id="31">
        <input>Schreibe eine Nachricht an Nummer 775 920 696
        mit dem Text Hallo wie geht es dir</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Hallo wie geht es dir</control>
    </sentence>
    <sentence id="32">
        <input>Schreibe eine Nachricht Hallo wie geht es dir
        an Nummer 775 920 696</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Hallo wie geht es dir</control>
    </sentence>
    <sentence id="33">
        <input>Schreibe Nachricht Hallo wie geht es dir
        an 775 920 696</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Hallo wie geht es dir</control>
    </sentence>
    <sentence id="34">
        <input>Schreibe Nachricht Hallo wie geht es dir an
        Nummer 775 920 696</input>
        <control>[APP]=Message;[PHONENUMBER]=775920696;
        [TEXT]=Hallo wie geht es dir</control>
    </sentence>
    <sentence id="40">
        <input>Schreibe eine Email mit dem Text Schicke Fotos
        aus dem Urlaub für Marta Graf</input>
        <control>[APP]=Email;[CONTACT]=[Graf Marta];
        [TEXT]=Schicke Fotos aus dem Urlaub</control>
    </sentence>
    <sentence id="41">
        <input>Schreibe eine Email an Thomas im Anhang schicke
        ich Konto</input>
        <control>[APP]=Email;[CONTACT]=[Müller Thomas, ...];
        [TEXT]=Im Anhang schicke ich Konto</control>
    </sentence>
```

```
<sentence id="42">
    <input>Email mit dem Inhalt schicke Fotos aus dem
    Urlaub für Renate</input>
    <control>[APP]=Email;[CONTACT]=[Graf Renate];
    [TEXT]=schicke Fotos aus dem Urlaub</control>
</sentence>
<sentence id="43">
    <input>Email mit dem Inhalt schicke Fotos aus dem
    Urlaub für Mama</input>
    <control>[APP]=Email;[CONTACT]=[Mama];
    [TEXT]=schicke Fotos aus dem Urlaub</control>
</sentence>
<sentence id="50">
    <input>Wie sieht das Wetter morgen aus</input>
    <control>[APP]=Weather;[DATE]=tomorow</control>
</sentence>
<sentence id="51">
    <input>Wetter heute</input>
    <control>[APP]=Weather;[DATE]=today</control>
</sentence>
<sentence id="53">
    <input>Wie wird morgen das Wetter sein</input>
    <control>[APP]=Weather;[DATE]=tomorow</control>
</sentence>
<sentence id="54">
    <input>Wie wird das Wetter am Montag</input>
    <control>[APP]=Weather;[DATE]=monday</control>
</sentence>
<sentence id="55">
    <input>Wie wird das Wetter am Dienstag</input>
    <control>[APP]=Weather;[DATE]=tuesday</control>
</sentence>
<sentence id="56">
    <input>Wie wird das Wetter am Mittwoch</input>
    <control>[APP]=Weather;[DATE]=wendsday</control>
</sentence>
<sentence id="57">
    <input>Wie wird das Wetter am Donnerstag</input>
    <control>[APP]=Weather;[DATE]=thursday</control>
</sentence>
<sentence id="58">
    <input>Wie wird das Wetter am Freitag</input>
    <control>[APP]=Weather;[DATE]=friday</control>
</sentence>
<sentence id="59">
    <input>Wie wird das Wetter am Samstag</input>
    <control>[APP]=Weather;[DATE]=saturday</control>
</sentence>
<sentence id="60">
    <input>Wie wird das Wetter am Sonntag</input>
    <control>[APP]=Weather;[DATE]=sunday</control>
</sentence>
<sentence id="61">
```

```
        <input>Stelle den Wecker auf 8:00</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="62">
        <input>Stelle den Wecker auf 16:00</input>
        <control>[APP]=Alarm;[TIME]=16:00</control>
    </sentence>
    <sentence id="63">
        <input>Wecke mich um 8:00</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="64">
        <input>Wecke mich um 8:00</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="65">
        <input>Wecke mich um 23:00</input>
        <control>[APP]=Alarm;[TIME]=23:00</control>
    </sentence>
    <sentence id="66">
        <input>Wecke mich um 8:00 vormittags</input>
        <control>[APP]=Alarm;[TIME]=8:00</control>
    </sentence>
    <sentence id="67">
        <input>Wecke mich um 8:00 nachmittags</input>
        <control>[APP]=Alarm;[TIME]=20:00</control>
    </sentence>
    <sentence id="68">
        <input>Wecke mich in 3 Stunden</input>
        <control>[APP]=Alarm;[TIME]=+3:00</control>
    </sentence>
    <sentence id="69">
        <input>Wecke mich in zwei Stunden</input>
        <control>[APP]=Alarm;[TIME]=+2:00</control>
    </sentence>
    <sentence id="70">
        <input>Stelle den Kurzzeitwecker auf eine Minute
        </input>
        <control>[APP]=Timer;[TIME]=1m</control>
    </sentence>
    <sentence id="71">
        <input>Stelle den Kurzzeitwecker auf zwei Minuten
        </input>
        <control>[APP]=Timer;[TIME]=2m</control>
    </sentence>
    <sentence id="72">
        <input>Stelle den Kurzzeitwecker auf drei Minuten
        </input>
        <control>[APP]=Timer;[TIME]=3m</control>
    </sentence>
    <sentence id="73">
        <input>Stelle den Kurzzeitwecker auf 3 Minuten
        </input>
```

```
        <control>[APP]=Timer;[TIME]=3m</control>
    </sentence>
    <sentence id="74">
        <input>Stelle den Kurzzeitwecker auf 4 Minuten
        </input>
        <control>[APP]=Timer;[TIME]=4m</control>
    </sentence>
    <sentence id="75">
        <input>Stelle den Kurzzeitwecker auf 5 Minuten
        </input>
        <control>[APP]=Timer;[TIME]=5m</control>
    </sentence>
    <sentence id="80">
        <input>Rechne 3 + 2 * 4 + 1 / 3 aus</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="81">
        <input>Rechne 3 plus 2 mal 4 plus 1 durch 3 aus
        </input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="82">
        <input>Berechne drei plus zwei mal vier
        plus eins durch drei</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="83">
        <input>drei plus zwei mal vier plus eins
        durch drei ausrechnen</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/3</control>
    </sentence>
    <sentence id="83">
        <input>drei plus zwei mal vier plus eins
        durch vier berechnen</input>
        <control>[APP]=Calc;[EXAMPLE]=3+2*4+1/4</control>
    </sentence>
    <sentence id="90">
        <input>Stelle eine Erinnerung auf Donnerstag
        um 8 Uhr ein</input>
        <control>[APP]=Calendar;[DATE]=Thursday;
        [TIME]=8:00</control>
    </sentence>
    <sentence id="91">
        <input>Stelle eine Erinnerung auf Donnerstag
        um 8 Uhr nachmittags ein</input>
        <control>[APP]=Calendar;[DATE]=Thursday;
        [TIME]=20:00</control>
    </sentence>
    <sentence id="92">
        <input>Erstelle eine Aufgabe für Samstag
        um 8 Uhr abends</input>
        <control>[APP]=Calendar;[DATE]=saturday;
        [TIME]=20:00</control>
```

```
    </sentence>
    <sentence id="93">
        <input>Füge ein Treffen in Kalender am
        Dienstag um 3 Uhr nachmittags ein</input>
        <control>[APP]=Calendar;[DATE]=tuesday;
        [TIME]=15:00</control>
    </sentence>
    <sentence id="94">
        <input>Einen Termin am Dienstag um 3 Uhr
        nachmittags in Kalender hinzufügen</input>
        <control>[APP]=Calendar;[DATE]=tuesday;
        [TIME]=15:00</control>
    </sentence>
    <sentence id="95">
        <input>Neuen Termin in Kalender am 12.4.2017
        um 11 Uhr vormittags hinzufügen</input>
        <control>[APP]=Calendar;[DATE]=2017-4-12;
        [TIME]=11:00</control>
    </sentence>
    <sentence id="96">
        <input>Füge in Kalender einen Termin am 18.4.
        um 9 Uhr früh ein</input>
        <control>[APP]=Calendar;[DATE]=2017-4-18;
        [TIME]=9:00</control>
    </sentence>
    <sentence id="97">
        <input>Füge in Kalender einen Termin am 31.5.
        um 10 Uhr vormittags ein</input>
        <control>[APP]=Calendar;[DATE]=2017-5-31;
        [TIME]=10:00</control>
    </sentence>
    <sentence id="98">
        <input>Einen Termin in Kalender am 31. April
        um 10 Uhr vormittags hinzufügen</input>
        <control>[APP]=Calendar;[DATE]=2017-4-31;
        [TIME]=10:00</control>
    </sentence>
    <sentence id="99">
        <input>Einen Termin in Kalender am 31. April 2018
        um 10 Uhr vormittags erstellen</input>
        <control>[APP]=Calendar;[DATE]=2018-4-31;
        [TIME]=10:00</control>
    </sentence>
    <sentence id="100">
        <input>Sage mir einen Witz</input>
        <control>[APP]=Joke</control>
    </sentence>
    <sentence id="101">
        <input>Erzähle einen Witz</input>
        <control>[APP]=Joke</control>
    </sentence>
    <sentence id="102">
        <input>Sage mir etwas lustiges</input>
```

```
        <control>[APP]=Joke</control>
    </sentence>
    <sentence id="103">
        <input>Erzähle mir eine Geschichte</input>
        <control>[APP]=Story</control>
    </sentence>
    <sentence id="104">
        <input>Lese mir eine Erzählung vor</input>
        <control>[APP]=Story</control>
    </sentence>
    <sentence id="105">
        <input>Sage mir eine Geschichte</input>
        <control>[APP]=Story</control>
    </sentence>
    <sentence id="106">
        <input>Erzähle mir ein Märchen</input>
        <control>[APP]=Fairy tales</control>
    </sentence>
    <sentence id="107">
        <input>Lese mir ein Märchen vor</input>
        <control>[APP]=Fairy tales</control>
    </sentence>
    <sentence id="110">
        <input>Öffne die Applikation Anrufen</input>
        <control>[APP]=Call</control>
    </sentence>
    <sentence id="111">
        <input>Öffne die Applikation Mitteilungen</input>
        <control>[APP]=Message</control>
    </sentence>
    <sentence id="112">
        <input>Öffne die Applikation Kontakte</input>
        <control>[APP]=Contacts</control>
    </sentence>
    <sentence id="113">
        <input>Öffne Einstellungen</input>
        <control>[APP]=Setting</control>
    </sentence>
    <sentence id="114">
        <input>Öffne Statusinformationen</input>
        <control>[APP]=Info</control>
    </sentence>
    <sentence id="115">
        <input>Öffne Favoriten</input>
        <control>[APP]=Favourites</control>
    </sentence>
    <sentence id="116">
        <input>Öffne MP3 Player</input>
        <control>[APP]=Player</control>
    </sentence>
    <sentence id="117">
        <input>Öffne Radio</input>
        <control>[APP]=Radio</control>
```

```
    </sentence>
    <sentence id="118">
        <input>Öffne Bücher</input>
        <control>[APP]=Books</control>
    </sentence>
    <sentence id="119">
        <input>Öffne Android Applikationen</input>
        <control>[APP]=AndroidApp</control>
    </sentence>
    <sentence id="120">
        <input>Öffne Farberkennung</input>
        <control>[APP]=ColorsRekognizer</control>
    </sentence>
    <sentence id="121">
        <input>Öffne Lupe</input>
        <control>[APP]=MagnifyingGlass</control>
    </sentence>
    <sentence id="122">
        <input>Öffne Notizen</input>
        <control>[APP]=Notes</control>
    </sentence>
    <sentence id="123">
        <input>Öffne Diktiergerät</input>
        <control>[APP]=Recorder</control>
    </sentence>
    <sentence id="124">
        <input>Öffne WLAN</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=OPEN</control>
    </sentence>
    <sentence id="125">
        <input>Öffne Banknotenerkennung</input>
        <control>[APP]=BankRecognizer</control>
    </sentence>
    <sentence id="126">
        <input>Handy ausschalten</input>
        <control>[APP]=ShutDown</control>
    </sentence>
    <sentence id="127">
        <input>Öffne Bedienungsanleitung</input>
        <control>[APP]=Manual</control>
    </sentence>
    <sentence id="130">
        <input>Schreibe eine Notiz mit dem Inhalt Milch
        Kaufen nicht vergessen</input>
        <control>[APP]=Notes;[TEXT]=Milch Kaufen nicht vergessen
        </control>
    </sentence>
    <sentence id="131">
        <input>Schreibe eine Notiz mit dem Text Milch
        Kaufen nicht vergessen</input>
        <control>[APP]=Notes;[TEXT]=Milch Kaufen nicht vergessen<
        /control>
    </sentence>
```

```
<sentence id="132">
    <input>Schreibe Notiz Milch Kaufen nicht vergessen</input>
    <control>[APP]=Notes;
    [TEXT]=Milch Kaufen nicht vergessen</control>
</sentence>
<sentence id="140">
    <input>Spiel Radio RADIO BOB! ab</input>
    <control>[APP]=Radio;[STATION]=RADIO BOB!</control>
</sentence>
<sentence id="141">
    <input>Schalte Radio LagoonFm ein</input>
    <control>[APP]=Radio;[STATION]=LagoonFM</control>
</sentence>
<sentence id="142">
    <input>Mache Radio Radio Regenbogen an</input>
    <control>[APP]=Radio;[STATION]=Radio Regenbogen</control>
</sentence>
<sentence id="150">
    <input>Spiele das Lied Crossroads</input>
    <control>[APP]=Player;[ARTIST]=The Offspring;
    [ALBUM]=The Offspring;[GENRE]=Alternative rock;
    [NAME]=Crossroads</control>
</sentence>
<sentence id="151">
    <input>Spiele irgendwas von Offspring</input>
    <control>[APP]=Player;[ARTIST]=The Offspring;
    [ALBUM]=The Offspring;</control>
</sentence>
<sentence id="152">
    <input>Spiele irgendein Pop ab</input>
    <control>[APP]=Player;[GENRE]=Pop</control>
</sentence>
<sentence id="153">
    <input>Mache das Lied Love Yourself an</input>
    <control>[APP]=Player;[ARTIST]=Justin Bieber;
    [ALBUM]=Purpose;[GENRE]=Pop;[NAME]=Love Yourself</control>
</sentence>
<sentence id="154">
    <input>Spiele das Album Songs of Innocence ab</input>
    <control>[APP]=Player;[ARTIST]=U2;
    [ALBUM]=Songs of Innocence</control>
</sentence>
<sentence id="160">
    <input>Akkustand</input>
    <control>[APP]=Phone;[STATUS]=BATTERY</control>
</sentence>
<sentence id="161">
    <input>Netzstärke</input>
    <control>[APP]=Phone;[STATUS]=SIGNAL</control>
</sentence>
<sentence id="162">
    <input>Akku</input>
    <control>[APP]=Phone;[STATUS]=BATTERY</control>
```

```
    </sentence>
    <sentence id="163">
        <input>Netzstärke</input>
        <control>[APP]=Phone;[STATUS]=SIGNAL</control>
    </sentence>
    <sentence id="164">
        <input>WLAN Stand</input>
        <control>[APP]=Phone;[STATUS]=WIFI</control>
    </sentence>
    <sentence id="165">
        <input>WLAN ausschalten</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=OFF</control>
    </sentence>
    <sentence id="166">
        <input>WLAN einschalten</input>
        <control>[APP]=Phone;[STATUS]=WIFI</control>
    </sentence>
    <sentence id="167">
        <input>WLAN an</input>
        <control>[APP]=Phone;[STATUS]=WIFI;[ACTION]=ON</control>
    </sentence>
    <sentence id="170">
        <input>Wie spät ist es?</input>
        <control>[APP]=Info;[CATEGORY]=TIME</control>
    </sentence>
    <sentence id="171">
        <input>Wie viel Uhr ist es?</input>
        <control>[APP]=Info;[CATEGORY]=TIME</control>
    </sentence>
    <sentence id="173">
        <input>Welches Datum ist heute?</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="174">
        <input>Welcher Tag ist heute?</input>
        <control>[APP]=Info;[CATEGORY]=DATE</control>
    </sentence>
    <sentence id="175">
        <input>Handy neustarten</input>
        <control>[APP]=Restart</control>
    </sentence>
    <sentence id="176">
        <input>Starte das Handy neu</input>
        <control>[APP]=Restart</control>
    </sentence>
    <sentence id="180">
        <input>Öffne die Applikation Facebook</input>
        <control>[APP]=Facebook</control>
    </sentence>
    <sentence id="181">
        <input>Öffne die Applikation TapTapSee</input>
        <control>[APP]=TapTapSee</control>
    </sentence>
```

```
    <sentence id="182">
        <input>Mache die App Facebook auf</input>
        <control>[APP]=Facebook</control>
    </sentence>
    <sentence id="183">
        <input>Öffne die Applikation WhatsApp</input>
        <control>[APP]=WhatsApp</control>
    </sentence>
    <sentence id="184">
        <input>Öffne Internet Browser</input>
        <control>[APP]=InternetBrowser</control>
    </sentence>
    <sentence id="185">
        <input>Öffne die Suchmaschine</input>
        <control>[APP]=InternetBrowser</control>
    </sentence>
    <sentence id="186">
        <input>Finde im Internet heraus wann Karl IV gestorben ist
        </input>
        <control>[APP]=InternetBrowser;[TEXT]=wann ist Karl IV
        gestorben</control>
    </sentence>
    <sentence id="187">
        <input>Suche wie hoch der Eiffelturm ist</input>
        <control>[APP]=InternetBrowser;[TEXT]=wie hoch ist der
        Eiffelturm</control>
    </sentence>
    <sentence id="188">
        <input>Finde wie hat gestern manchester united gespielt</input>
        <control>[APP]=InternetBrowser;[TEXT]=wie hat gestern
        manchester united gespielt</control>
    </sentence>
</resources>
```