

Óbudai Egyetem
Neumann János Informatikai Kar
Informatikai Rendszerek Intézete



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

**NYELVTANULÁS INTELLIGENS MOBIL
ÁGENSSEL**

Szerző: **Pásztor Tibor Viktor**
mérnök informatikus szak, IV. évf.

Konzulens: **Dr. Kutor László**
egyetemi docens

Budapest, 2012.

NYELVTANULÁS INTELLIGENS MOBIL ÁGENSSSEL

Pásztor Tibor Viktor

Óbudai Egyetem

Neumann János Informatikai Kar, IV. évfolyam

Konzulens: Dr. Kutor László, egyetemi docens

Magyarországon alapvető társadalmi problémának tekinthető az idegen nyelvek ismeretének széleskörű hiánya. A nyelvtanulóknak sok esetben nincs lehetősége a beszélt nyelvet elegendő mélységben gyakorolni, csupán könyvekből, a nyelvtani szabályok elsajátításával próbálnak tanulni. Ez azonban sajnos a legtöbb esetben nem hoz kielégítő eredményeket, így a tanulók könnyen feladhatják a további tanulást.

Az évek során rengeteg próbálkozás született az élő, beszélt nyelv tanításának számítógépes támogatására, ám a hangfelismerő és hangszintetizáló technológiák egészen eddig nem érték el a megfelelő fejlettségi szintet ahhoz, hogy ezek a megoldások tényleg hatékonyak lehessenek. Manapság azonban ezek a területek rohamos fejlődésen estek át az okostelefonok elterjedésének következtében.

A TDK projektem célja egy olyan intelligens ágens kialakítása, amely a modern hangfelismerő és hangszintetizáló technológiákat felhasználva, élő nyelven folyó, adaptív párbeszédet képes lebonyolítani a felhasználóval, fejlesztve annak beszédképességét és szókincsét. Ezen párbeszédet a programban a manapság népszerű „chat” alkalmazások formájában jelennek meg.

A program minden leckéhez külön tudásbázisban rögzíti a tanítandó kifejezéseket, azok egymástól való függését és a felhasználó eddig elért eredményeit. Ezen ismeretek alapján progresszív tanulási folyamat építhető fel a felhasználó új ismeretekkel való túlterhelése nélkül.

A TDK folyamán a projekt az angol nyelv oktatására fókuszál, de a későbbiekben nem kizárt annak kibővítése további nyelvekre is.

Tartalomjegyzék

Bevezetés.....	6
Miért van szükség egy ilyen eszközre?	6
Hogyan lehet ezt megvalósítani?.....	6
Mit tervezek a projektben megvalósítani?.....	7
Miért most jött el ennek az ideje?	7
Irodalomkutatás	8
Az okostelefonokon jelenleg elérhető nyelvtanuló alkalmazások	8
uTalk English (Android/iPhone) [4]	8
Oscar Emanuel - Advanced English Vocabulary (Android) [5]	9
Brainglass – Read4English (Android/iPhone) [6].....	10
Nyelvtanuló módszerek.....	12
The Michel Thomas Method [7]	12
Pimsleur Language Programs [8].....	12
Hangfelismerés (Speech-to-text).....	13
Google SpeechRecognizer API.....	14
Nuance Dragon Speech [12]	14
CMU Sphinx - Speech Recognition Toolkit [15].....	15
Összegzés	15
Hangszintetizáció (Text-to-speech).....	15
Nuance Dragon Speech [12]	16
Google Translate	16
SVOX [16]	16
Összegzés	17
Rendszerterv	17
Bevezetés.....	17
Lecke adatbázis	18

Szerkezet (függőségi gráf)	18
Tárolás	20
Automatizálhatóság lehetősége (vizuális lecketervező).....	21
Működési terv.....	21
Felhasználási esetek	21
A továbblépés irányának az eldöntése	23
Egyéb tárolt adatok.....	24
Implementáció	25
Implementációs platform.....	25
Modulok és kapcsolataik	25
Felhasználói felület	26
Text-to-Speech	27
Speech-to-Text	28
Tanár.....	29
Adatbázis vezérlő (Leckevezérlő).....	30
Tesztelés	30
Tesztelési terv.....	30
Speech-to-Text tesztek	30
Text-to-Speech tesztek	32
Internethasználat tesztek	33
Kliens tesztek	34
Összegzés	35
Mi készült el?	35
Fejlesztési lehetőségek	36
Tartalmi fejlesztési lehetőségek	36
Funkcionális fejlesztési lehetőségek	37
Üzleti fejlesztési lehetőségek	37

Ábrajegyzék	39
Irodalomjegyzék.....	39

Bevezetés

Miért van szükség egy ilyen eszközre?

Az idegen nyelvek ismeretének fontosságát nem kell senkinek sem bizonyítani. És aki valaha tanult már nyelvet, az azt is tudja, hogy hatékonyan nyelvet tanulni a megfelelő eszközök nélkül nagyon nehéz. Ilyen „eszközök” alatt érthetjük a szótárakat, a szótanuló gyakorlatokat, a nyelvtani összefoglalókat, nyelvtankönyveket, nyelvtanárokat, és az idegen nyelvi beszélgetőpartnereket is.

Az hogy valakinek ezen eszközök mely kombinációja válik be, azt személye válogatja, de az nem vitás, hogy a szóbeli gyakorlás minden esetben elengedhetetlen a siker elérésének érdekében. Ennek az oka természetes és magától értetődő, hiszen általában élő, beszélt nyelveket szeretnénk elsajátítani.

A nyelvtanulók nagy többsége viszont már itt problémákba ütközik, aminek vagy az az oka, hogy nem találnak maguknak megfelelő beszélgetőpartnert, vagy sokan csak a gátlásaikból adódóan nem mernek mások előtt megszólalni, szégyellve saját hiányos nyelvtudásukat (pedig nincsen abban semmi szégyellnivaló). Az itt körvonalozódó igény kielégítésére jöhetnek képbe a számítógéppel támogatott interaktív nyelvoktató eszközök.

Hogyan lehet ezt megvalósítani?

A szóbeli gyakorlás automatizálásának rengeteg megvalósítása létezik már jelenleg is. Legegyszerűbb esetben gondoljunk azokra az audio kazettákra, amelyeken felolvassák nekünk a leckét, amire mi a sípszó után válaszolhatunk. Ezek után elhangzik a válasz, aminek a segítségével ellenőrizhetjük a tudásunkat.

Léteznek különböző szövegértésen alapuló gyakorlatok is, ahol egy hallott szöveget kell értelmeznünk, és a hallottak alapján kell kérdésekre válaszolnunk.

A fejlődés egy következő lépcsőfoka a hangfelismerő algoritmusokat alkalmazó nyelvtanuló alkalmazások. Ezek jelenleg főleg csak a kiejtés helyességére koncentrálnak, a kiejtett szöveg hangképének mintához való illesztésével. Az 1. ábrán egy ilyen program látható működés közben.



1. ábra – A Tell Me More [1] nevű hangfelismerő algoritmust alkalmazó nyelvtanuló alkalmazás. Ez csak a kiejtés helyességére koncentrál, a kiejtett szöveg hangképének egy mintához való hasonlításával.

Látható, hogy már ez a három módszer is teljesen más tanulási területet céloz meg.

Mit tervezek a projektben megvalósítani?

A TDK projekt célja egy olyan intelligens ágens kialakítása, amely a modern hangfelismerő és hangszintetizáló technológiákat felhasználva, élő nyelven folyó, adaptív párbeszédet képes lebonyolítani a felhasználóval, fejlesztve annak beszédképességét és szókincsét. Ezen párbeszéd a programban a manapság népszerű „chat” alkalmazások formájában jelennek meg.

Miért most jött el ennek az ideje?

A hangfelismerők a kiejtés vizsgálatánál komolyabb célokra való használatának eddig technikai akadályai voltak. Egészen eddig ezek az algoritmusok nem működtek elég megbízhatóan ahhoz, hogy egy nyelvtanuló amúgy is bizonytalan kiejtését felismerjék, azt szöveggé alakítsák.

Az okostelefonok elterjedésének köszönhetően a hangfelismerő, és hangszintetizáló eszközök rohamos fejlődésen mentek keresztül, amelynek hatásait a napjainkban megjelenő, hangfelismerésen alapuló kontextus alapú keresőszolgáltatásokat nyújtó alkalmazásokon is

megfigyelhetünk. Ilyen alkalmazások az egy éve megjelent Apple Siri-je [2], illetve az idén nyáron megjelent Google Now [3] is.

Irodalomkutatás

Az okostelefonokon jelenleg elérhető nyelvtanuló alkalmazások

Az irodalomkutatás első lépéseként elengedhetetlen felmérni a jelenlegi piaci helyzetet az alkalmazás célterületén. Ennek a legegyszerűbb és legcélravezetőbb módja, a jelenleg elérhető alkalmazások kipróbálása, és ezekből a tanulságok leszűrése.

A vizsgálat az alábbi szempontokra fókuszál:

- Tanított terület
- Tanítási módszer
- Interaktivitás
- Adaptivitás
- Tanított nyelvek
- Felhasználói felület

uTalk English (Android/iPhone) [4]

Ez egyértelműen egy szótanuló alkalmazás. Tanuló módban a magyar szavakat egy listában találhatjuk meg, azokra rákattintva megjelenik azok képe, angol megfelelője, illetve elhangzik a szó kiejtve is. Gyakorló módban egy több nehézségi fokkal rendelkező memóriajátékon keresztül van lehetőségünk a tudásunk ellenőrzésére, illetve elmélyítésére. A szavakat a program kategóriákra bontva tartalmazza, amit a 2. ábrán is láthatunk. A tudásunkat a program pontozással értékeli, a végső cél a maximális pontszám elérése minden kategóriában. Az alkalmazás jellegéből adódóan könnyen fordítható, így rengeteg nyelven elérhető. A felhasználói felület könnyen kezelhető, modern, tetszetős, szépen animált.



2. ábra – Az uTalk alkalmazás kategórianézete.

Összegzés: A program legnagyobb pozitívumának a felhasználói felülete írható fel, amelynek köszönhetően nagyon kellemes a használata. Negatívum: A legnehezebb nehézségi fokozat hibátlan elvégzése egyáltalán nem a felhasználó nyelvtudását teszi próbára, csupán a rövid távú memóriáját. Ezáltal az alkalmazás pár óra alatt végigjátszható anélkül is, hogy értenénk mit csinálunk.

Oscar Emanuel - Advanced English Vocabulary (Android) [5]

Szintén egy szótanító alkalmazással van dolgunk. Ennek a megközelítése azonban teljesen más, mint az előzőé, annál sokkal játékosabb. Az egész alkalmazás egy játék jellegét adja, pályákkal, pályánkénti csillagos pontozással és teljes képernyős grafikákkal, amiket a 3. ábrán láthatunk. A játék menete egyszerű. Angolul kapunk utasításokat hogy keressünk meg tárgyakat az adott pályán. Azokat megtalálva plusz pontot kapunk, rossz tárgyak megérintéséért viszont pontlevonást jár. Rendelkezőnk négy sűgóval is, amik megmutatják a keresett tárgy helyét. A program csak angolul jelenít meg információkat, így bármilyen visszajelzés a szavak jelentéséről csak képeken át történik.



3. ábra - Advanced English Vocabulary pályaválasztója. Ebből is jól látszik az alkalmazás játékos jellege.

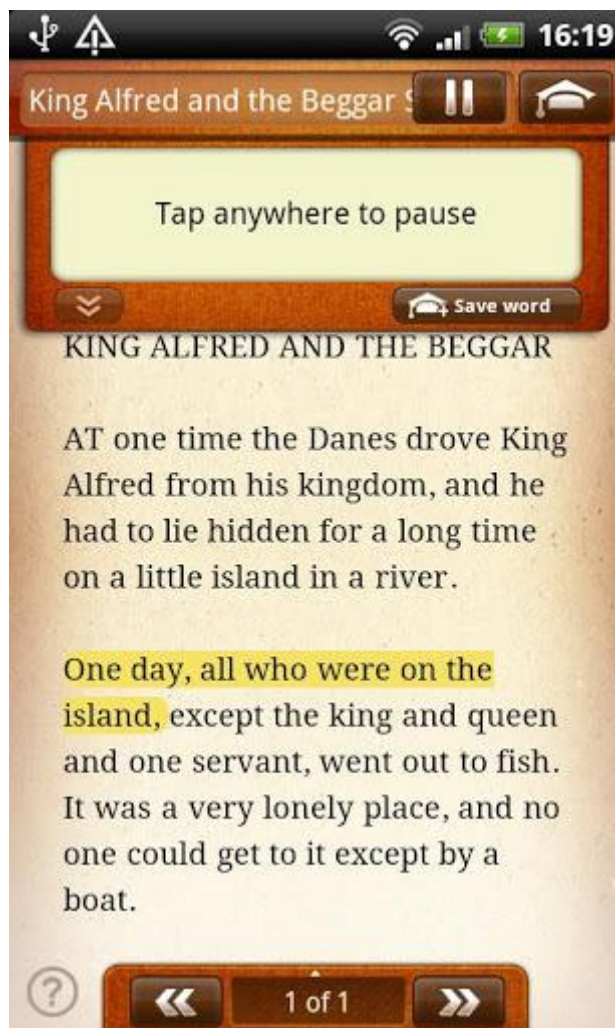
Összegzés: Pozitívumok: Játékos, tetszetős, gyerekeket is könnyen lekötő felület. Negatívum: Nem tartalmaz semmiféle tanító metódust a sűgókon kívül, szóval aki nem ismer egy kérdezett szót arra súlyos pontlevonással járó próbálgatás vár.

Brainglass – Read4English (Android/iPhone) [6]

Érdekes, hallott szöveg értésén alapuló alkalmazásról van szó. A program próbaverziójában elkezd lejátszani JF Kennedy „Inaugural Address” című beszédét, amit írásban is megjelenít, és abban szóról szóra kiemeli az épp elhangzó részt. Pont mintha az egy karaoke program lenne. Innen ered az eredeti címe is, amely Karaoke4English volt. Erről mutat egy pillanatképet a 4. ábra.

A szöveget lehetőségünk van bármikor megállítani, ilyenkor bármelyik szóra kattintva megjelenik annak jelentése, illetve elrakhatjuk azt saját szótárunkba. Miután végighallgattuk a leckét, lehetőségünk van a menet közben félrerakott szavainkat forgatható kártyákon át

memorizálni, illetve begyakorolni. Ez a funkció az önellenőrzés elvén működik, szóval miután felfordítottunk egy kártyát, nekünk kell visszajelezni, hogy jól tudtuk e, vagy sem az adott szót, és a program eszerint értékeli minket is.



4. ábra - A Read4English alkalmazás "karaoke" jellegű felülete.

Lehetőségünk van az alkalmazáson belül további szövegeket is letölteni, azonban ezek már pénzbe kerülnek. Viszont pozitívum, hogy ezeket közvetlenül az alkalmazáson belül lehetőségünk van egy gomb nyomására megvásárolni, és azok így rögtön használhatóvá is válnak. A program angol nyelvet tanít az alábbi nyelv beszélői számára: ukrán, olasz, orosz, koreai, portugál, lengyel, japán, kínai, indonéz, maláj, francia, holland, arab, spanyol, német, török. Ezek között sajnos a magyar nem található meg, viszont a megvalósítás minősége miatt előnyösnek tartottam mégis ezen alkalmazás kielemezését.

Összegzés: Pozitívum: Intuitív módon köti össze a hallott és az írott szöveget a nyelvtanuló számára. Az alkalmazáson belüli vásárlás megoldása elegáns.

Nyelvtanuló módszerek

Mivel egy nyelvoktató programról van szó, fontos megvizsgálni egy pár nyelvtanulási módszert is, amikből ötleteket meríthetünk a működést illetően.

Az évek során két nyelvkurzussal találkoztam, amelyek a szóbeliségen, és a természetes, beszélgetés alapú nyelvtanuláson alapulnak. Ezek a Michel Thomas, és a Pimsleur módszer. Mind a kettő rendkívül hatékonynak bizonyult, és meglepően hasonlítanak egymásra. Viszont sajnos egyik sem érhető el magyar nyelven. Alapvetően angoloknak készültek, és számtalan európai, és egyéb nyelvet lehet velük tanulni.

Bizonyos szinten ezek a nyelvkurzusok ihlették meg az egész alkalmazást is, azzal a különbséggel, hogy most az angol nyelv oktatásán van a hangsúly magyar nyelvtanulók számára, illetve az egész tanítást egy intelligens alkalmazás végezné, különböző leckéken át.

The Michel Thomas Method [7]

Személyesen volt szerencsém végighallgatni a módszer teljes német audio nyelvkurzusát, illetve kipróbálni a francia kezdő kurzusát.

Az alábbiakban a weblapjukon található ismertetőt foglalom össze.

A módszer alapmottója a gyorsan elért magabiztosság a nyelvben, könyvek, jegyzetelés, és erőltetett memorizálás nélkül. A mondatok építőelemeikre bontásán alapul, így azok kombinálásával könnyedén tudjuk felépíteni a saját mondatainkat is. A tanulási lépések egymásra épülnek, és a megértésen alapulnak. A hanganyagokban a nyelvtanár mellett két tanuló is részt vesz, így a kurzusok egy virtuális tanterem hangulatát adják, ahol mi vagyunk a harmadik diák. Ezek a tanulók éppen ugyan olyan kezdők a nyelvben, mint mi magunk, így osztozhatunk a sikerükben, illetve tanulhatunk a hibáikból is. A tanítás, az emlékezés, vagyis az emlékeztetés feladata teljes mértékben a tanáré. Így a tanulók felszabadultan, stressz nélkül sajátíthatják el az új ismereteket.

Pimsleur Language Programs [8]

Ehhez a módszerhez is volt szerencsém, a kezdő dán nyelvkurzusukon keresztül.

Szintén a hivatalos weblapjuk ismertetőjét összefoglalva:

A Pimsleur módszer szerint a tanulás kulcsa a kérdések megfelelő feltételében van. Olyan kérdések feltevésével, amelyek felidéznek a tanulóban a válasz nyelvtani szerkezetét, ösztönös válaszokat lehet előidézni.

A módszer négy elven alapszik:

1. Fokozatosan ritkuló ismétlések: Dr. Pimsleur kutatásai az emberi memória és tanulás területén arra a következtetésre vitték, hogy ha egy tanulót megadott, egyre növekvő időközönként emlékeztetünk egy új szóra vagy kifejezésre, az egyre hosszabb ideig képes azt megjegyezni. Így könnyedén a rövid távúból a hosszú távú memóriába lehet juttatni a tanulandó kifejezéseket.
2. Az ösztönösség elve: Amikor már valaki jól beszél egy nyelvet, és kérdeznek tőle valamit, a válasz szinte már ösztönösen jön, gondolkodás nélkül. Erre a szintre eljutni persze idő és rengeteg munka. Viszont a diákokban már a korai stádiumban előidézhető ez a fajta viselkedésmód megfelelő ingerléssel, ezzel is gyorsítva a tanulás menetét.
3. Kötődés az alapszókinchez: Különböző kutatások [9] kimutatták, hogy az általános mindennapi interakciók a szókinész csak egy meglehetősen szűk készletét használják. Ebből ered, hogy a tanulás korai fázisaiban felesleges túlzottan nagy szókinészt a diákra zúdítani, azzal csak a tanulási folyamatot lassítjuk le. Később, viszont ez a minimális szókinész már megfelelő alapot biztosít annak önálló magabiztos növelésére.
4. A természetes tanulás elve: A módszer a funkcionális tanulást részesíti előnyben, a megértésre és beszédre alapozva a kezdetektől kezdve.

Hangfelismerés (Speech-to-text)

Az alkalmazás lelkét a leckéken és a módszertanon túl, a hangfelismerő eszköz nyújtja. Több ilyen eszköz érhető el a piacon, így fontos megfontolni melyiket választjuk az alkalmazásunkhoz.

A vizsgálat az alábbi szempontokra fókuszál:

- Találati pontosság magyar nyelven
- Találati pontosság angol nyelven
- Találati pontosság rövid kifejezéseknél
- Találati pontosság hosszabb mondatok esetén
- Találati pontosság hibás mondatok esetén
- Félrehallás mértéke
- Dokumentáció, támogatottság
- Licenzelhetőség

- Offline működés

Google SpeechRecognizer API

Mivel első lépésben Android platformra tervezem a fejlesztést, ez az eszköz tűnik a természetes választásnak, viszont csupán erre alapozni a döntést nem szabad.

Ezt az eszközt lehetőségem volt kipróbálni több, ezt használó alkalmazáson keresztül is. Ezek viszont annyira újak voltak, hogy első lépésként szükséges volt a telefonomon egy szoftverfrissítést végezni, a jelenleg legfrissebb, 4.1-es Android verzióra. [10]

A vizsgálatot a Google Voice Search [11] szolgáltatás kipróbálásával kezdtem. A tapasztalat sajnos az volt, hogy magyar nyelven az eszköz teljesen használhatatlan. Nagyon ritkán találta el azt, amit mondok, még egyszerű kifejezéseknél is. Angol nyelvre állítva viszont az eredmények meglepően jók voltak. Az eszköz kimondottan pontosnak bizonyult rövid és hosszú kifejezéseknél is. Külön pozitívumként nyilvánult meg, hogy a nyelvtanilag hibás mondatokat is pontosan felismerte az esetek többségében. Fontos azt is megvizsgálni, hogy az esetleges félrehallások esetén mennyire tér el a felismert szöveg az elhangzottól. Ebben az esetben is jó volt a helyzet. Mivel az eszköz része az Android SDK-nak, így annak a dokumentáltságával, támogatottságával, és licenszelhetőségével sincs probléma.

Az eszköz egy jelentős hátránya, hogy aktív internetkapcsolatot igényel a működéséhez, ugyanis a tényleges számításokat egy a felhőben elhelyezkedő szerver végzi.

Nuance Dragon Speech [12]

A Nuance Dragon Speech az az eszköz, amely Apple Siri-jét hajtja. Ennek létezik mobil SDK-ja, amely elérhető iPhone és Android platformon is. A vizsgálatot a „Dragon Go!” [13], és az ezt használó Swype Beta [14] alkalmazás kipróbálásával végeztem. Magyar nyelven sajnos itt sem értem el túlzott sikereket, angol nyelven viszont már jobbak voltak az eredmények. A találati arány meglehetősen jó volt rövid, és elfogadható volt hosszabb kifejezések esetén is. Az egyetlen problémát a hibás és félrehallott mondatok jelentették, ugyanis ilyen esetekben a rendszer néha olyan eredményeket adott vissza, amelyeknek köze sem volt az eredeti kifejezésekhez. Dokumentáció és támogatás bőven rendelkezésre áll, licenz szempontjából több lehetőség is elérhető az ingyenessétől, a több ezer dolláros arany csomagig.

Előnyként írható fel hogy a felismerés offline módban is működik.

CMU Sphinx - Speech Recognition Toolkit [15]

A CMU Sphinx egy nyílt forráskódú, hangfelismeréssel foglalkozó egyetemi projekt, amelyet a Carnegie Mellon egyetem fejleszt. Ez az eszköz egy rendkívül rugalmas és jól konfigurálható megoldást biztosít a hangfelismerésre, viszont használata mélyebb ismeretek igényel a témában. Az oldalukon elérhető demók alapján arra a következtetésre jutottam, hogy ezen eszköz esetében a Nuance vagy Google termékeivel vetélkedő funkcionalitás eléréséhez rengeteg munkát kell még befektetni, ami nem feltétlenül a jelen projektem célja. Így a továbbiakban inkább az előbb tárgyalt két eszköz felhasználási lehetőségeire koncentrálok.

Összegzés

A leszűrt eredményeket egy táblázatban foglalom össze, a könnyű áttekinthetőség érdekében.

	Google SpeechRecognizer	Nuance Dragon Speech
Találat pontosság magyar nyelven	Rossz	Rossz
Találati pontosság angol nyelven	Jó	Jó
Találati pontosság rövid kifejezéseknél	Jó	Jó
Találati pontosság hosszabb mondatok esetén	Jó	Közepes
Találati pontosság hibás mondatok esetén	Jó	Rossz
Félrehallás mértéke	Csekély	Jelentős
Dokumentáció, támogatottság	Jó	Jó
Offline működés	Nincs	Van

Mivel jelenleg a fő hangsúly a találatok, és főleg a hibás találatok pontosságán van, így a választás a Google hangfelismerőjére esett.

Hangszintetizáció (Text-to-speech)

A hangfelismerés után a hangszintetizáció a második legfontosabb külső eleme az alkalmazásomnak. Erre szintén rengeteg eszköz áll rendelkezésre, viszont ezeknek csak egy

korlátozott részhalmaza támogatja a magyar nyelvet is. Így a keresést már rögtön az elején le is szűkíthetjük.

A vizsgálat az alábbi szempontokra fókuszál:

- Magyar nyelvi támogatás
- Angol nyelvi támogatás
- Hangsúlyozás, természetes hangzás
- Beszédhibák
- Offline működés
- Licenzelhetőség

Nuance Dragon Speech [12]

A Nuance SDK részét képezi egy hangsintetizáló eszköz is. Ez támogatja a magyar és az angol nyelvet egyaránt. A szintetizáló természetes hangsúlyozást alkalmaz, aminek köszönhetően nincs robohangja a beszédhangoknak. Sajnos azért minden hangmintán felismerhetőek voltak rendszeresen visszatérő beszédhibák, amikből egyértelműen kiderült, hogy egy hangsintetizáló szoftverről van szó. Az eszköz online működésű, szóval egy felhőből érkeznek a felolvasott kifejezéseknek megfelelő hangfájlok. A licenzelhetőségi lehetőségek azonosak a Nuance hangfelismerőjével, hiszen azonos szoftvercsomagba tartozik a két eszköz.

Google Translate

A Google Translate rendelkezik egy felolvasó webszolgáltatással, amely egyaránt támogatja a magyar és az angol nyelvet is, természetes hangú szintetizációval. Az angol kiejtése rendkívül természetes, beszédhibákkal alig rendelkezik. Magyar nyelven kicsit tagolt, néha természetellenesen hangsúlyozott, de az esetek nagy többségében nem rossz. Mivel ez egy webszolgáltatás, a felolvasott kifejezések itt is a felhőből jönnek vissza hangfájlok formájában. A szolgáltatás ingyenesen használható.

SVOX [16]

A SVOX beépülő hangsintetizáló szolgáltatást nyújt Android operációs rendszert futtató telefonokra. Ezt a szolgáltatást más alkalmazások is elérhetik, de csak akkor, ha a megfelelő SVOX csomagok telepítve vannak a készülékre. A szolgáltatás licenzelhető saját alkalmazásba is, de csak olyan módon, hogy a saját alkalmazásunkon belül felkérjük a felhasználót, hogy telepítse a megfelelő SVOX csomagokat, amiket azután mi használni

tudunk. Ez nem csak, hogy feleslegesen komplikálja a felhasználók életét, de ezek a csomagok fizetősek is. Így Az alkalmazásunk rögtön azzal nyílna, hogy meg akar vetetni két fizetős alkalmazást a felhasználójával. Ez sajnos megengedhetetlen, még akkor is, ha egyébként a felajánlott hangszintetizálási szolgáltatások igen magas színvonalúak.

Egyébként a cég hivatalos oldalának a linkje [17], ma már egyenesen a korábban tárgyalt Nuance cég oldalára visz, szóval valószínűleg a felvásárlás esete állhat fent.

Összegzés

	Google Translate	Nuance Dragon Speech	SVOX
Magyar nyelvi támogatás	Van	Van	Van
Angol nyelvi támogatás	Van	Van	Van
Hangsúlyozás, természetes hangzás	Jó	Jó	Jó
Beszédhibák	Ritkák	Előfordulnak	Előfordulnak
Offline működés	Nincs	Nincs	Van
Licenzelhetőség	Ingyenesen	Jó	Nem megfelelő

A döntés fő szempontjai most is a nyelvi helyességen vannak, így a döntés a Google Translate hangszintetizáló szolgáltatására jutott.

Rendszerterv

Bevezetés

Összegezve az irodalomkutatásban tanultakat a nyelvtanulási módszertanokról, egy hatékony szóbeli nyelvtanuló alkalmazásban ügyelni kell az új ismeretek fokozatos bevezetésére, a tanuló túlterhelése nélkül. Ezen felül a megvizsgált alkalmazások alapján kiderült, hogy érdemes arra is figyelni, hogy csak olyan dolgokat kérjünk számon a felhasználtól, amit már meg is tanítottunk neki.

Következő lépésben gondolni kell a leckék tényleges menetére. Itt jön képbe az, amit a Pimsleur módszernél fedeztünk fel, az ismétlések gyakoriságáról. Tehát érdemes az újonnan

tanult kifejezéseket eleinte többet, majd idővel egyre ritkábban és ritkábban ismételtetni. Persze ez a módszer csak akkor tartható, ha működik is. Szóval folyamatosan rögzíteni kell a felhasználó visszajelzéseit, és tanulási tempóban alkalmazkodni kell a tanuló képességeihez.

Lecke adatbázis

Az előző gondolatokból alakult ki egy függőségeken alapuló tudásbázis kép, amik később a tényleges leckéket fogják alkotni.

Szerkezet (függőségi gráf)

Tehát a tanítandó kifejezéseket érdemes lenne egy függőségi gráfban tárolni. Így könnyű lenne kiválasztani azokat a kifejezéseket, amikhez például minden szükséges szót megtanítottunk már, tehát amelyeknek nincsen kielégítetlen függőségük.

Fontos kérdés hogy mely függőségeket tároljuk. Mindet, vagy csak a közvetleneket.

Teljes függőségi gráf (rekurzív lekérdezések)

Az alapötlet az lenne, hogy tároljunk el minden függőséget. Ez esetben egy teljes függőségi gráfról beszélünk. Ezeknek a létrehozása egyszerű, viszont a lekérdezések folyamán minden függőséget le kellene vizsgálnunk, amihez rekurzív lekérdezésekre lenne szükségünk.

Felmerül ezzel egy másik probléma is. A nyelvtanulási módszertanoknál arra is rájöttünk, hogy érdemes a tanuló számára bevezetett szóincset eleinte minimálisan tartani, és azokat is egymáshoz kötődő relációban átadni. Így azokból később, mint meglévő építőkövekből tudunk majd újabb és újabb kifejezéseket alkotni.

Tehát semmiképpen sem teljesen véletlenszerűen szeretnénk a nyílt függőségekkel rendelkező kifejezéseink között haladni, hanem valamiféle logika szerint.

Szükségszerűvé válik tehát egy logikai sorrendiséget, vagy függőségi rendszert is tárolni a már meglévő függőségi gráfunkon belül. Ez túlzottan is elbonyolítaná a rendszert, képlékennyé, nehezen kezelhetővé tenné a leckék elkészítését. Ezért egyszerűbb és egyben célravezetőbb szerkezetet kellene választanunk.

Irányított fa (folyamatorientáltabb)

Szükséges e minden függőséget eltárolnunk? Hogyha a tanulandó kifejezések egyszerre vannak függőségi kapcsolatban egymással, és egyszerre alkotnak egyfajta logikai sorrendet is,

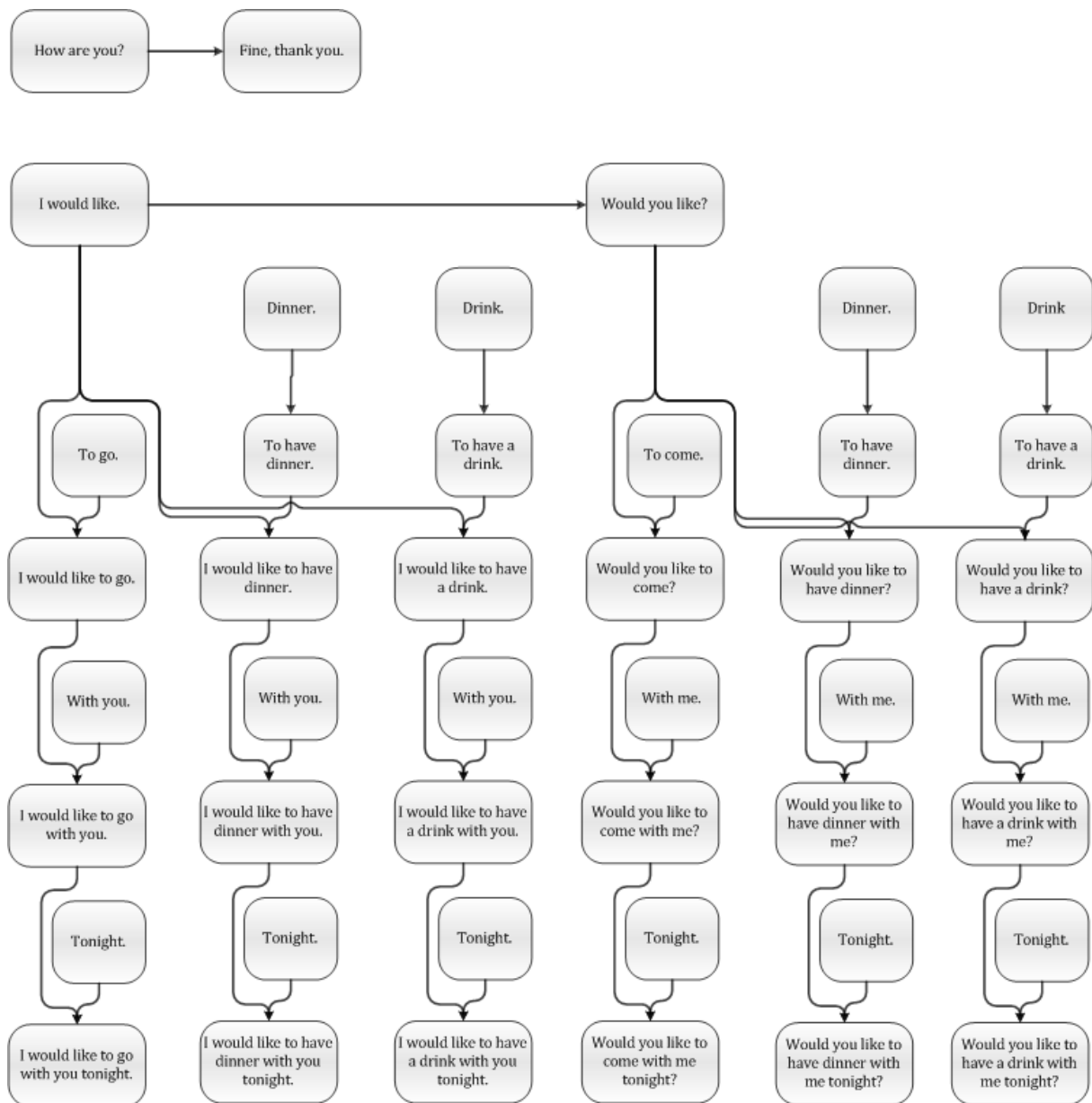
amely logikai sorrend függ a függőségi sorrendtől, adja magát a tanulás sorrendjét egy fa szerkezetbe szervezni.

A kifejezések függőségei között él a tranzitivitás elve, így felesleges minden függőséget eltárolnunk. Elég a közvetlen rákövetkező függéseket tárolni, mint szülő és gyerekkifejezések.

Ezen felül így kontrollálni tudjuk az új szókincs bevezetését is azzal, hogy csak akkor hozunk be új szavakat, amikor a fában ott tartunk, tehát tényleg szükség van rájuk egy új tanítandó kifejezéshez.

Ebben a szerkezetben a lekérdezéseink is egyszerűsödnek, ugyanis a helyes működést és a konzisztens állapotot feltételezve (vagyis feltételezve hogy egészen eddig nem ugráltunk a fában, hanem csak a kielégített függőségű kifejezések irányában haladtunk tovább), elég most már csak egy kifejezés szüleit vizsgálni, annak eldöntése érdekében, hogy annak vannak e kielégítetlen függőségei.

Ezek alapján nézzük meg egy példa lecke gráfját az 5. ábrán. Az ábra átláthatóságának érdekében némelyik pont többszörözve jelenik meg, de ez a rákövetkezők konzisztenciáját nem rontja el, természetesen a kész adatbázisban csak egyszer fognak szerepelni.

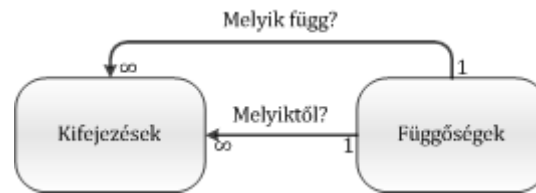


5. ábra – A példalecke irányított függőségi gráfja. Az ábra egyszerűsítésének érdekében egy kifejezés több pontként is megjelenhet.

Tárolás

Az elkészült példa leckénket ezek után szeretnénk letárolni valamilyen számítógépes adatstruktúrába. Több szerkezet és eszköz áll a rendelkezésünkre erre a célra, mégis a legcélravezetőbb és a legkönnyebben kezelhető egy adatbázis lenne hozzá.

Egy ilyen adatstruktúra tárolására alkalmas adatbázis szerkezeti rajzát a 6. ábrán láthatjuk.



6. ábra – A függőségek tárolásának szerkezeti rajza egy adatbázisban.

Felmerül a kérdés milyen adatokat tároljunk az egyes táblákban. Ez a működés tervezése során derül majd ki részletesebben.

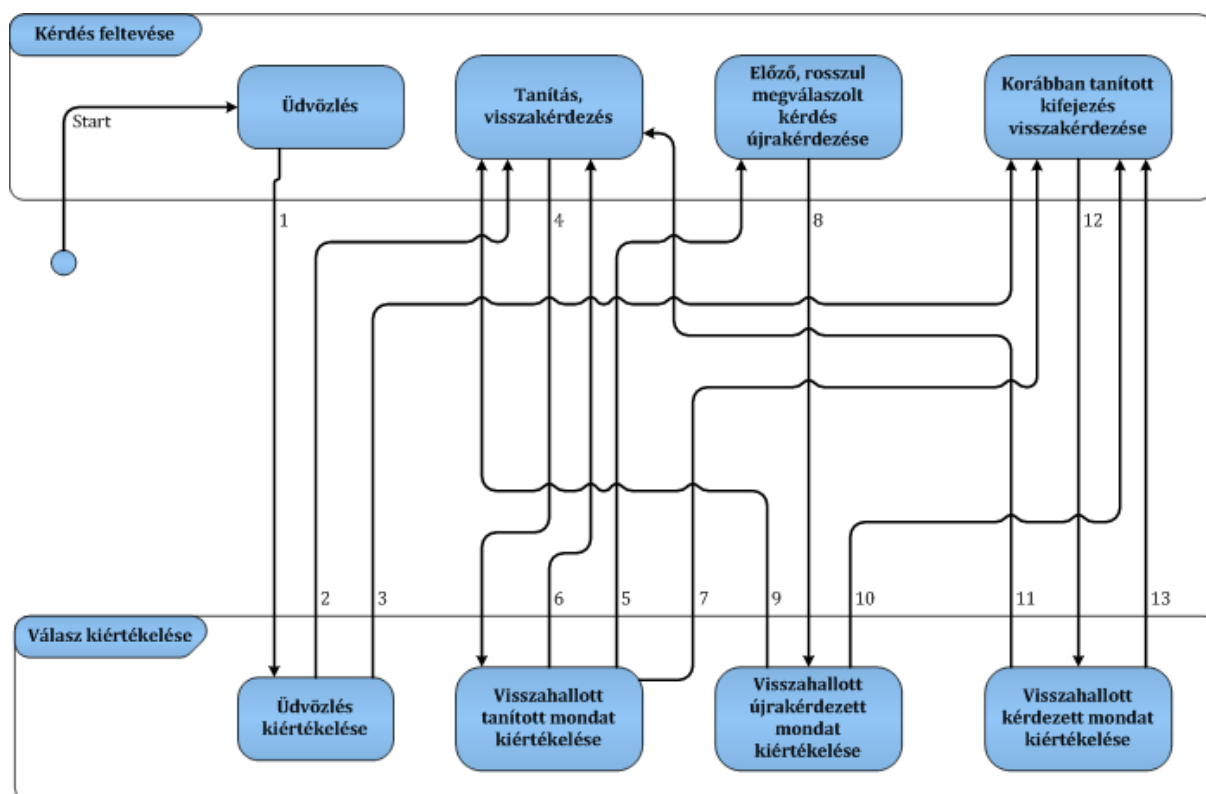
Automatizálhatóság lehetősége (vizuális lecketervező)

Már most látszik, hogy egy ilyen lecke kézi elkészítése és adatbázisba szervezése macerás feladat. Viszont egy vizuális szerkesztővel könnyen létrehozható lenne, amiből már a kész adatbázis automatikusan legenerálható. Ez mindenképpen a projekt egy potenciális továbbfejlesztési lehetősége.

Működési terv

Felhasználási esetek

Alapvetően három jelentős működési eset különíthető el. Az üdvözlés, a tanítás, az ismétlés, és a kérdezés. Ezeket legjobban egy állapot-diagrammal lehet szemléltetni, amit a 7. ábrán láthatunk.



Állapotátmenetet kiváltó esemény:

1. A felhasználó üdvözlése.
2. Az üdvözlést kiértékeljük, és nincs több visszakérdezendő kifejezés, új tanítására van szükség.
3. Az üdvözlést kiértékeljük, és van visszakérdezendő tanított kifejezés.
4. A felhasználó megismételte a frissen tanult kifejezést.
5. A felhasználó rosszul ismételte meg a frissen tanult kifejezést.
6. A felhasználó sikeresen megismételte a frissen tanult kifejezést. Ezek után nincs több visszakérdezendő kifejezés, új tanítására van szükség
7. A felhasználó sikeresen megismételte a frissen tanult kifejezést. Ezek után van visszakérdezendő tanított kifejezés
8. A felhasználó újra megismételte az előzőleg kért kifejezést.
9. A felhasználó sikeresen megismételte az előzőleg kért kifejezést. Ezek után nincs több visszakérdezendő kifejezés, új tanítására van szükség
10. A felhasználó sikeresen megismételte az előzőleg kért kifejezést. Ezek után van visszakérdezendő tanított kifejezés.
11. A felhasználó megválaszolta a feltett kérdést. Ezek után nincs több visszakérdezendő kifejezés, új tanítására van szükség
12. A felhasználó megválaszolta a feltett kérdést.
13. A felhasználó megválaszolta a feltett kérdést. Ezek után van visszakérdezendő tanított kifejezés.

7. ábra - A program állapotdiagramja.

Első ránézésre ez az ábra bonyolultnak tűnhet, de lényegében annyiról van szó, hogy a program kérdez, amire a felhasználó válaszol. A kérdések három fajtájúak lehetnek. Vagy egy új, most tanult kifejezést ismételt meg, vagy egy régebben tanult kifejezést kér vissza, vagy az előzőleg elrontott kifejezést ismételteti meg. Ezekre a felhasználó válaszol, aminek

hatására, attól függően, hogy jó vagy rossz válasz érkezett, a program új állapotba lép, és újra kérdez. Vegyük kicsit részletesebben ezeket az állapotokat.

Üdvözlés

A program indulásakor üdvözlí a felhasználót, ezután egy válaszra vár. Ha megfelelő üdvözlés jött vissza válaszul, akkor továbbléphetünk a tanulásra vagy a kérdezésre. Arra a döntésre, hogy melyikre, még később kitérünk. Ha viszont nem megfelelő válasz érkezett, a program elmagyarázza, hogyan kell angolul üdvözölni valakit.

Tanítás

Tanításról beszélünk, ha olyan kifejezést vezetünk be, amiről korábban még nem volt szó. Ilyen esetben elhangzik az angol kifejezés, annak a magyar megfelelője, majd az alkalmazás felkér minket annak megismétlésére. Ha sikeresen megismételtük a kifejezést, akkor továbbléphetünk, ha nem, a program átugrik az ismétlés állapotába.

Ismétlés

Ismétlés állapotában a program felszólítja a felhasználóját, hogy ismétlje meg az előzőleg elrontott kifejezést. Helyes válasz esetén továbbléphetünk, rossz válasz esetén a program állapota stagnál.

Kérdezés

Kérdezés állapotában a program olyan kifejezést kér vissza tőlünk, amit korábban már tanított. Ilyen esetben adhat segítséget is. A válasz beérkezése után, akár jó, akár hibás válasz érkezett, az eredményt naplózzuk, és továbblépünk.

A továbblépés irányának az eldöntése

A korábbiakban mindig csak annyit mondtunk, ha a felhasználó helyesen válaszolt, továbblépünk vagy egy kérdésre, vagy egy új tanítandó kifejezésre. Viszont annak az eldöntése, hogy ezek közül ténylegesen melyikre, külön feladatot alkot.

Ennek támogatására bevezetjük a kifejezések pontozásának a rendszerét. Minden kifejezésnél tárolásra kerülnek az alábbi adatok:

- **Tanítva:** A kifejezés hányszor lett tanítva.
- **Kérdezve:** A kifejezés hányszor lett megkérdezve a felhasználótól.

- **Megválaszolva:** A kérdések során hányszor válaszolt helyesen a felhasználó. Ez az érték értelemszerűen kisebb vagy egyenlő, mint a kérdezve adattag értéke.
- **Tanítás ideje:** A kifejezést hányadikként vezettük be a tanulónak, mint új kifejezés.

Ezek segítségével definiálhatjuk az alábbi kifejezéseket is:

- **Tanítatlan kifejezés:** Azokat a kifejezéseket nevezzük tanítatlannak, amelyeknél a „tanítva” adattag értéke 0.
- **Kielégített függőségű kifejezés:** Azoknak a kifejezéseknek kielégített minden függősége, amelyek vagy nem rendelkeznek függőségekkel, vagy minden függőségük „tanítva” adattagja nagyobb, mint 0.
- Érdemes felvennünk a már megtanított kifejezések számát is mint változó. Nevezzük ezt az **aktív kifejezések számának**.

Ezek segítségével az alábbi szabálykészletet állíthatjuk fel:

- Egy kifejezést csak akkor kérdezhetünk, ha már azt megtanítottuk.
- Új kifejezést akkor tanítunk, hogyha minden eddig megtanult kifejezést már megfelelően begyakoroltunk.
- Egy kifejezést megfelelően betanítottunk, hogyha a bevezetése óta minden új kifejezés bevezetése után legalább egyszer megisméltettünk. Vegyük észre hogy ezzel a módszerrel, a kifejezések számának növekedésével arányosan nő az ismétlések között eltelt idő is. Ez az irodalomkutatásban megismert fokozatosan ritkuló ismétlések módszere szerint ideális.
- Egy korábban nem tanított kifejezést akkor vezethetünk be, ha az kielégített függőségű.

Vegyük észre, hogy ez a szabálykészlet valamilyen szinten önkényes, a későbbiekben leszűrt tapasztalatok és teszteredmények alapján még változhat.

Egyéb tárolt adatok

Az előző pontozási rendszeren felül lehetőségünk van még számtalan egyéb, hasznos információ eltárolására egy kifejezéshez az adatbázisban. Ilyenek:

- Alternatív, kifejezéshez kapcsolódó kérdés és válaszmódok.
- Szó szerinti fordítások, a szórend könnyebb megjegyzésének érdekében.
- Kifejezéshez kapcsolódó egyedi magyarázatok.

- Sikeres válasz vagy tanítás után alternatív, kifejezéshez kapcsolódó dicséret, jutalmazások.

Implementáció

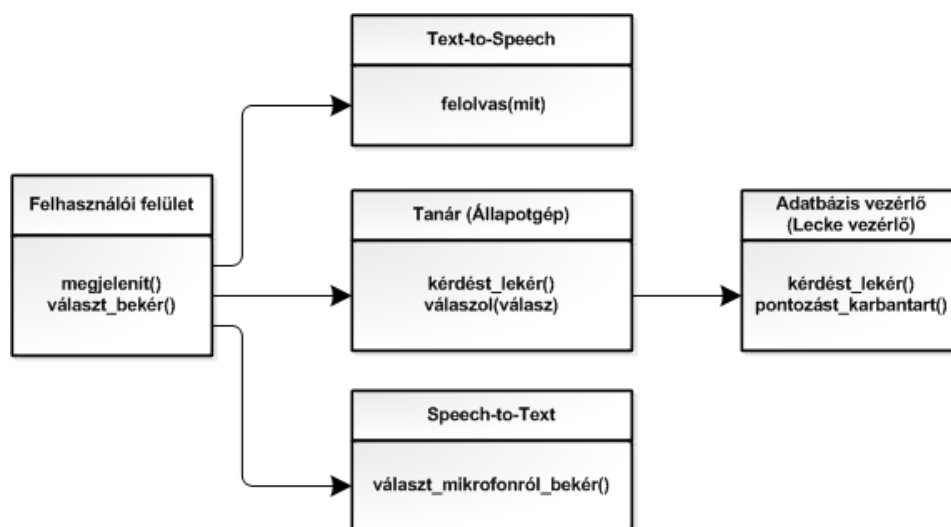
Implementációs platform

A korábbiakban már több utalás történt az Android platform használatára, viszont a fejlesztés kezdetével fontos ezt a döntést ténylegesen is megerősíteni. Ami a platformok képességei alapján akár szóba is jöhetnek az az Android és az iPhone. Az Android rendszer melletti döntésem többrétű:

- Az irodalomkutatás során eldöntésre került a Google hangfelismerő és hangsintetizáló eszközeinek a használata. Ezeknek a támogatottsága Android rendszer alatt igen jó. iPhone rendszer alatt valószínűleg a Nuance termékei lennének a célravezetőbbek, bár a Google eszközök is rendelkeznek külső webes interfészekkel.
- Piacilag mind a két platform megfelelő célpontnak minősül, így idővel a kész alkalmazás portolásra is kerülhet. A legjobb az lenne, ha végül mind a két platformon elérhető lenne.
- Eszközellátottsággal, és fejlesztési tapasztalattal is jobb helyzetben állok az Android területén.

Modulok és kapcsolataik

Az eddigiek alapján a programot a 8. ábrán látható fő modulok fogják alkotni. A diagramon látható azok logikai kapcsolatai is.



8. ábra – A program fő moduljai.

Felhasználói felület

A felhasználói felület kialakításánál alapvető szempontok voltak:

- Célravezető, jól átlátható felület
- Könnyű kezelhetőség
- Ízléses megjelenés

Az alapötlet egy egyszerű csevegőalkalmazás szerű megjelenés volt. Ennek a mostanság divatos módja a szövegbuborékok használata. A korai fázisban az alkalmazásom is ezt a megjelenést követte. Ez látható a 9. ábrán. A zöld buborékok a felhasználó mondatai, a szürkék és a fehérek a telefon válasza.



9. ábra – Képernyőkép az alkalmazás korai megjelenéséről. Itt a manapság népszerű szövegbuborékos csevegőstílust alkalmaztam. A zöld buborékok a felhasználó mondatai, a szürkék és a fehérek a telefon válasza.

Ezek után jött az a designötlet, hogy az alkalmazás egy régi nyelvkönyv hangulatát adja klasszikus párbeszédleírásokkal, és lapozási effektusokkal. Ennek a megvalósítása látható a 10. ábrán. A vörös mondatok a felhasználótól származnak, a feketék és a barnák a telefon válasza.



10. ábra – Képernyőkép az alkalmazás új designjáról. Ebben egy régi könyv formáját veszi fel, klasszikus párbeszédleírásokkal, és lapozási effektusokkal. A vörös mondatok a felhasználótól származnak, barnák és feketék a telefon válasza.

Az ábrán két dolog figyelhető meg. A főmenüben körvonalozódnak az elérhető legfontosabb funkciók, illetve a chat ablakban fellelhető egy „fej” ikon a sarokban. Azt megérintve aktivizálódik a hangfelismerő, és kezd el figyelni arra, amit mondunk. Ez a gomb természetesen csak akkor aktív, ha a gép épp nem beszél.

Text-to-Speech

A hangszintetizáló osztály lényegében egy webszolgáltatást használ, így ennek a legfőbb feladatai az alábbiak:

- szöveg előfeldolgozás
- letöltések kezelése
- gyorsítótár építés és tisztántartás
- a visszakapott hangfájlok lejátszása

A működése az alábbiak szerint alakul. Kap egy kifejezést, amit fel kell olvasnia. Ezt a kifejezést megtisztítja a felesleges szóközöktől és írásjelektől, a gyorsítótár szennyezés

elkerülése érdekében. Ezek után megvizsgálja, hogy a kifejezés szerepel-e már a kifejezés gyorsítótárban. Ha igen, akkor lejátssza onnan azt, ha nem, akkor letölti, gyorsítótárazza, és ezután játssza le.

Ehhez a működéshez több észrevételt is lehet fűzni. Az egyik az, hogy az egy leckéhez kapcsolódó felolvasandó hanganyagok száma, a felhasználó válaszait leszámítva, véges. Tehát felesleges készülékenként ennyi kéréssel bombázni a kiszolgálószervert, ezek előre letölthetők lehetnének egy csomagban, a leckével együtt.

A másik az, hogy a leckék folyamán nagyon gyakoriak az egymásra épülő, összetett felolvasandó mondatok. Vegyül az alábbi példát. A pár oldallal előbb található 9. ábrán az alábbi szövegbuborék látható:

„Ez magyarul azt jelenti, hogy: Hogy vagy? Próbáld meg kimondani.”

Ezt a program első körben három részben olvassa fel, tagmondatonként. De amikor már kimondásra került a teljes mondat, és a felhasználó azt újra felolvastatja, az már, mint egy kifejezés jelenik meg. Szét lehetne ezeket tárolási szinten is bontani, de az jelentősen megbonyolítaná a párbeszédet kezelő algoritmust. A szép megoldás az lenne, ha ezt a felolvasó modul észlelné, és bontaná fel futási időben, a többi modul tudta nélkül. Ez egy lehetséges, és idővel szükséges továbbfejlesztési lehetősége a rendszernek.

Speech-to-Text

A szövegfelismerő osztály szintén webszolgáltatást használ, viszont ezt már keretrendszeren keresztül teszi.

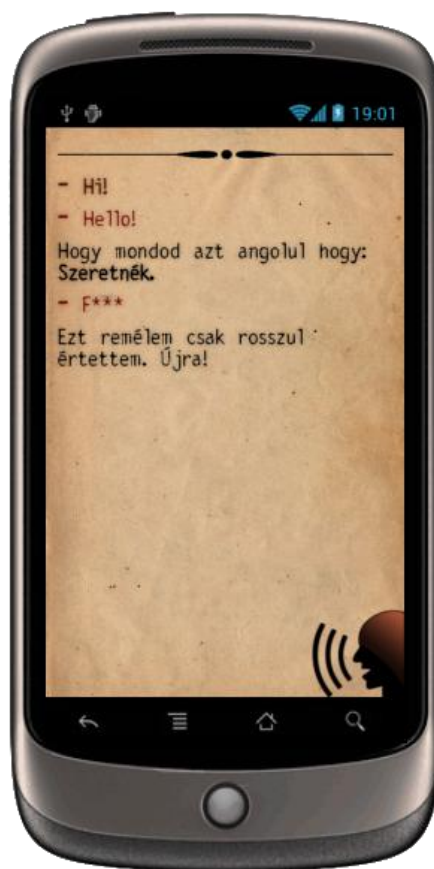
Ennek a feladatkörébe tartozik:

- A felhasználó beszédének a rögzítése
- A felvett anyag szerverre küldése, előforduló hibák lekezelése
- A szerver válaszána a feldolgozása
- Az esetleges nemkívánatos eredmények kiszűrése

Mik lehetnek nemkívánatos eredmények? A legfontosabb példák erre a trágár kifejezések. Ha még a felhasználó tényleg trágárságokat is beszélt bele a felismerőbe, akkor sem illik azokat megjeleníteni, sőt mi több vissza is olvasni. A helyzet még rosszabb, hogyha ezek felismerési hiba miatt csúsztak bele a találatok listájába.

A Google hangfelismerő API-ja rendelkezik erre beépített lehetőséggel, viszont az csak rendszerszinten kapcsolható be, kódból nem. Szóval ennek használatára az lenne az egyetlen lehetőség, hogy a felhasználót elnavigáljuk a telefon beállításainak a megfelelő menüjébe, és megkérjük, hogy pipálja be az adott opciókat. Ez nem egy elegáns megoldás.

A saját megoldásban klasszikus, minta alapú szűrést alkalmazok ahol, ha egy kifejezésben bárhol megtalálható egy, a tiltólistán lévő szavak közül, akkor az egész kifejezést eltávolítjuk. Előfordulhat az az eset is, hogy minden visszakapott eredményt eltávolítottunk. Ilyenkor nagyon nagy a valószínűsége, hogy a felhasználó tényleg trágárságot mondott, ezért ezt intelligensen kell lekezelni, majd a választ újra kell mondatni. Ez látható a 11. ábrán.



11. ábra – Képernyőkép a trágár kifejezések szűrésének egy lehetséges, intelligens módjáról.

Tanár

A tanár képviseli a korábban megtervezett működésű állapotgépet. Feladatai:

- Kommentálja a tanulás menetét, kérdez, verbálisan értékeli és tanít. Ezt egy kategóriákba rendezett kifejezés-tudásbázis segítségével végzi.
- Ellenőrzi a felhasználótól visszakapott válaszokat.

- Megfelelően lépked az állapotai között.
- Igény esetén új kérdést vagy tanítandó kifejezést kér le a leckevezérlőtől.

Adatbázis vezérlő (Leckevezérlő)

Az alkalmazás a népszerű MVC tervezési minta [18] ismeretében készült. Ezek alapján a leckevezérlő lényegében az adatbázis modelljeként fogható fel, amelynek kizárólagos hozzáférése van a fizikai adatbázishoz. Így az alkalmazás bármely másik része csak rajta keresztül érhet el adatokat az adatbázisból.

Tesztelés

Az alkalmazás tesztelése nem egy helyen és időben történt, több ember végezte több eszközön, a fejlesztés teljes időtartama alatt. Ezért itt csak a legfontosabb tesztelési lépéseket, és azok eredményeit fogom összefoglalni.

Tesztelési terv

Mivel az alkalmazás alapvetően modulos szerkezetű, lehetőségünk van a modulokat külön-külön saját különleges tulajdonságaik szerint tesztelni.

- A hangfelismerő modul tesztelésének kritikus pontja a hangfelismerés minőségének a vizsgálata különböző körülmények között.
- A hangsintetizátornál alapvető szempont a szintetizáció minőségének a vizsgálata, viszont itt különös figyelmet kell fordítani arra is, hogy mivel ez egy webszolgáltatás, ez idővel változhat. Fontos így az alkalmazásunk alkalmazkodási képességének a tesztelése ebből a szempontból.
- Mivel az előző két modul mind a kettő webszolgáltatáson alapszik, fontos tesztelni a rendszerek működését különböző minőségű internetkapcsolatok használata esetén is.
- A modulok egyedi tesztjei után szükséges a rendszeren egységtesztet is végezni, vizsgálva az általános működést, és a modulok optimális együttműködését.

Speech-to-Text tesztek

A Google hangfelismerő egy nagyon új eszköz, ami napi szintű fejlesztés alatt áll. Szóval könnyen megeshet, hogy a most felfedezett gyengeségei hamarosan javításra vagy továbbfejlesztésre kerülnek. Ettől függetlenül fontos a felismerő jelenlegi képességeinek a felmérése megfelelő tesztek elvégzésével, az alkalmazásom kontextusában.

Fontos volt felmérni a felismerés pontosságát,

- különböző típusú mondatokon
- különböző kiejtésű és nyelvi szinten álló embereknél
- különböző artikulációval beszélő embereknél
- különböző környezetekben

Az ember azt gondolná, hogy egy mondat bonyolódásával nő a felismerés hibáinak is a száma. Ez valamilyen szinten igaz, viszont minél több információ van egy mondatban, azt annál könnyebb pontosan beazonosítani. Ennek köszönhetően a felismerő megbízhatóan működött hosszabb mondatok esetén is. Akkor kezdtek problémák adódni, amikor túl rövid kifejezéseket adtunk be neki. Ez belegondolva logikus is, hiszen ha csak egyetlen egy kifejezés áll a felismerő rendelkezésére, vegyük példának a „foot” szót, azt rossz kiejtés esetén nagyon könnyű félreérteni akár az alábbi szavak egyikének is: „food”, vagy „fool”. Ezek alapján figyelni kell arra, hogy a leckékben egy bizonyos kifejezés-komplexitás alá ne menjünk, inkább tanítsunk összetettebb kifejezések útján.

A különböző kiejtésű és nyelvi szinten álló embereknél is az elvárásokhoz képest ellentétes eredmények születtek. Az angol nyelvben teljesen kezdő embereknél, akik rendszeren artikuláltak, és ténylegesen az alkalmazás által megadott hangmintát próbálták utánozni, a felismerő kimagasló pontossággal működött. Problémák az olyan, angol nyelvben magasabb szinten álló embereknél jelentkeztek, akik hadarva, az évek során kialakult akcentusukkal beszélték a nyelvet. De itt sem volt jelentős a probléma. Kicsit lelassítva, jobban odafigyelve a tiszta beszédre már jól működött a rendszer náluk is. Alapvetően nem is ők az alkalmazás célközönsége, szóval ezt a problémát nem nevezném jelentősnek.

Az alkalmazás tesztelésre került különböző csendesebb és zajosabb környezetekben is. Ehhez hozzá kell tenni, hogy mivel a felvételek az Android keretrendszeren keresztül történnek, automatikusan működésbe lépnek a telefon hardveres és szoftveres dinamikus zajszűrő és frekvenciaszűrő eszközei.

Elmondható, hogy egy nyugodt, csendes szobában, a program kényelmesen használható, akár asztalra letett telefonról is.

A helyzeten nem rontott sokat a szobában, akár közvetlen a telefon mellett szóló hangos zene sem. A felismerés olyan hangos zene mellett is jól működött, hogy a telefon válaszát már nem

hallottuk, ő viszont a miénket, megfelelően közel hajolva beszéd közben, igen. Az is mindegy volt, hogy hangszeres, vagy énekes zenéről van szó.

A készülékek igen jól kiszűrrik a statikus zajokat is, mint az utca zaja, vagy akár az egy közlekedési eszközön történő használat zaja.

Problémák olyan helyen jelentkeztek csak főleg, ahol a zaj forrása emberek beszélgetéséből származott a környezetünkben. Viszont ilyen esetben is ritkán fordult csak elő az, hogy a felismerő belehallott volna valamit a mi mondatunkba, egy szomszédos párbeszédből. Inkább volt jellemző az, hogy nem ismert fel semmit, így újrapróbálkozhattunk. Ebben a környezetben a megoldást az jelentette, hogy a telefonhoz közel hajolva beszélünk. Ilyenkor az esetek jelentős részében már működött rendesen a rendszer.

Text-to-Speech tesztek

A hangszintetizáló modulban többértű tesztekre volt szükség. Egyszer, webszolgáltatás révén, fontos kitérni az internetes elérési időkre és az adathasználatra. Erre külön fejezetet szánok, koncentráljunk most a tényleges hangszintetizációs képességekre.

Az egyes kifejezések felolvasása során felmerült az igény azok szavankénti felbontására, és külön felolvasására. Ennek tesztelése során merült fel az a probléma, hogy amíg egy teljes mondatot kimondottan szép kiejtéssel olvas fel a szintetizáló, egy-egy különálló szóra ez már nem mindig mondható el. Például ha beadunk neki egy „are” létigét egy mondatban, például „We are here.”, nem tapasztalunk problémákat. Viszont ha beadjuk neki az „are” létigét egymagában, az eredmény egy elnyelt torokhang. Pontosan olyan, mint amilyennek a mondat közepén hangzik dinamikus beszéd közben. Erre az eredményre oda kell figyelni a leckék kialakítása, és a mondatok felbontása közben, a hangszintetizációs gyorsítótár számára.

Szerencsés dolog viszont hogy ez a Google szolgáltatás is viszonylag friss, és manapság igen népszerű, így szintén napi szintű frissítéseket kap. Ennek következtében fordult elő az az eset is, hogy az egyik nap tesztelt, hibásan hangszó szót, pár nappal később már megfelelően ejtette ki az alkalmazás.

Bekövetkezett viszont egy másik, nem várt esemény is. Az egyik napi szintű frissítés során egy az egyben kicserélték a webszolgáltatás hangszintetizáló motorját. Ennek eredményeként a következő gyorsítótár ürítés után, a korábban megszokott női, enyhén robothatangú, de egyébként szépen beszélő hang helyett, egy sokkal szebb és természetesebb hangzású, viszont

férőhang várt minket. Ez alapvetően egy pozitív változás, viszont nem várt komplikációkkal járt.

A férfiak hangja általánosan alacsonyabb frekvenciájú, vagyis mélyebb, mint a nőké. A telefonok hangszórójáról viszont tudni kell, hogy méretükből adódóan sokkal jobban kiadják a magasabb hangokat, mint a mélyeket. Szóval az eredetileg azonos hangerejű férfi és női hangokat telefonon lejátszva azt tapasztalhattuk, hogy a férfihangok sokkal halkabban szólnak a nőieknél. Ezt a programon belül megfelelően korrigálni kellett.

Ebből az esetből leszűrhető tanulságként a gyorsítótárazás, és a lecke előletöltés fontossága is. Ezen technikák segítségével ugyanis minimalizálható a program webszolgáltatástól való függése.

Internethasználat tesztek

Az előzőleg tesztelt mindkét modul aktív internetkapcsolatot igényel, ezért fontos felmérni azok hogyan viselkednek különböző minőségű internetkapcsolatokon való használat esetén.

Ezen felül érdemes megnézni a generált adatforgalmukat is, hiszen előfordulhat, hogy az alkalmazást korlátos mobilinternet csomagokról fogják használni.

Lassú, 2G alapú internetkapcsolat esetén a rendszer még elég reszponzívnek bizonyult, viszont már észrevehetőek voltak a csatlakozási idők. Tehát érdemes a tényt, hogy most webes lekérés történik, a felhasználóval megfelelően közölni, hogy ne érezze úgy, hogy megfagyott az alkalmazás.

Adathasználati szempontból az alábbi táblázatban láthatunk egy-egy reprezentatív mérési eredményt:

Felolvasott angol kifejezés	Elhasznált adatmennyiség	Felolvasott magyar kifejezés	Elhasznált adatmennyiség
to come	3 744 bájt	jönni	5 538 bájt
Would you like?	5 760 bájt	Szeretnél?	7 210 bájt
Would you like to come?	7 056 bájt	Szeretnél jönni?	8 255 bájt
		Tanuljunk új kifejezést!	12 539 bájt
		Nagyon jó! A helyes válasz:	14 838 bájt

Látható hogy egy-egy kifejezés adathasználata átlagosan olyan 3-15 KB között van. Így egy 100 kifejezésből álló adatgyorsítótár felépítése körülbelül 1 megabájtnyi adatforgalmat generál. Ez tűrhető értéken belül van. Viszont még így is érdemes előletölteni, mert egy 1 megabájtos leckeméretet bőven megér az, hogy utána a szintetizáció már csak minimális további adatforgalmat generáljon.

Ezek után vizsgáljuk meg a hangfelismerő által generált adatforgalmat is. Erre segítségül használjuk a telefon Android 4.0.4-es verziójától elérhető beépített adatforgalom mérőjét. Ezen mérések eredményét az 12. ábrán láthatjuk 10, és 35 kifejezés felismertetése után. Látható hogy egy kifejezés átlagosan 1 KB adatforgalmat használ fel. Ez az érték teljes mértékben megengedhető.



12. ábra – Az Android rendszer beépített adatforgalom mérőjének állásai a mérések után. A képernyőképek 10, és 35 felismert kifejezés után készültek. Látható hogy egy kifejezés átlagosan 1 KB adatforgalmat használ.

Kliens tesztek

A modulok speciális tesztjein kívül szükség volt még az alkalmazást egységesen is tesztelni, működésének szempontjából.

Az első ilyen tesztek során előkerülő észrevétel az volt, hogy a telefon kommentárjai idővel redundánsak, unalmasak lesznek. Ennek a kiküszöbölésére született egy olyan adatbázis, ahol a megfelelő kommentárokhoz alternatívákat tárolunk el, amiből a program véletlenszerűen válogat. Viszont ez a megoldás a problémát csak enyhíti, de nem szünteti meg. A tényleges megoldást a kifejezéseként tárolt alternatív kérdések és válaszok jelentették, amelyek nem csak színessé teszik a párbeszédet, de a hatékony asszociatív tanulást is elősegítik.

Másik ilyen észrevétel az volt, hogy a leckék során feltett kérdésekre a választ a felhasználó könnyedén visszanézheti, felgörgetve a párbeszéd előzményeihez. Ezt úgy lehet kiküszöbölni, hogy minden kérdés előtt töröljük a képernyőt, egy elegáns lapozási effektussal.

A harmadik észrevétel az volt, hogy a leckék menete során a program ugyan jól kezelte a függőségeket, mégis kicsit túl nagy volt a tanított kifejezések szórása. Nem jött teljesen át azok egymásra épülése, és az ebből származó tanulási előnyök. Szóval a feltételeknek megfelelő kifejezések közül nem elég véletlenszerűen válogatni. Azok között definiálni kell a leckében egy ajánlott sorrendet is, amit ha nem is kőkeményen, de jó közelítéssel követni kell. Ennek kezelése a vizuális leckeszerkesztő feladata lesz.

Összegzés

A projekt eredetileg definiált céljai az alábbiak voltak:

„A TDK projektem célja egy olyan intelligens ágens kialakítása, amely a modern hangfelismerő és hangszintetizáló technológiákat felhasználva, élő nyelven folyó, adaptív párbeszédet képes lebonyolítani a felhasználóval, fejlesztve annak beszédképességét és szókincsét. Ezen párbeszédet a programban a manapság népszerű „chat” alkalmazások formájában jelennek meg.”

Mi készült el?

Sikeresen elkészült az alkalmazás magjaként szolgáló tudásbázis szerkezet, amely minden szükséges információt tartalmaz a leckék logikus, inkrementális felépítéséhez.

Ezen váz alapján elkészült egy mintalecke is, amelynek főleg a tesztelésben volt, és a demózásban lesz szerepe. A leckék tartalmi szerkezetének kialakítására még külön munkát kell majd fordítani, de ennek megoldása már nem informatikai probléma.

Ezen felül elkészült a párbeszédet kezelő teljes felhasználói felület és logika is, így a próbalecke alapján már teljes párbeszédet lehet lebonyolítani az alkalmazással. Eközben az alkalmazás folyamatosan rögzíti a felhasználó teljesítményét, és ezen adatok alapján alkalmazkodik a tanuló szintjéhez, a leckében való haladási sebesség meghatározásában.

Ugyan ahhoz, hogy az alkalmazásból kész, eladható piaci termék legyen, szükséges még pár kiegészítő funkció hozzáadása, viszont elmondható, hogy sikerült egy olyan új, egyedi nyelvtanító eszközt létrehozni, amely hatékonyan segítheti a nyelvtanulók életét a modern technológia, és nyelvoktató módszertanok segítségével.

Fejlesztési lehetőségek

Tartalmi fejlesztési lehetőségek

Vizuális leckeszerkesztő

Az adatbázis tervezését és megvalósítását az a megjegyzés zárta, hogy a leckék elkészítésének a folyamatát rendkívül meggyorsítaná egy vizuális szerkesztőeszköz kialakítása, amelyben könnyen megadhatók és szemléltethetők a kifejezések egymástól való függései. Az elkészült leckéket az eszköz végül kész, programba illeszthető adatbázisként tudná exportálni. A tényleges leckék felépítésének az első lépése mindenképpen egy ilyen eszköz megvalósítása kell hogy legyen.

Többnyelvűség

A bevezetőben szóba jött az, hogy a program idővel nem csak az angol nyelv oktatására lehet alkalmas, hanem más nyelvekére is. Ehhez az alkalmazásban nincs is szükség jelentős módosításokra, szóval érdemes lenne egy külön beépített mechanizmust létrehozni ennek a kezelésére, még akkor is, ha jelenleg még nincs rá szükség.

Leckék menetének a befolyásolása

A kliens működésének tesztelése során felmerült az, hogy hasznos lenne a leckéken belül, a kifejezések bevezetési sorrendjét, egy extra adattaggal ajánlás szinten meghatározni, hogy a tanulás menete jobban kövesse az egymásra épülés logikáját. Ennek a hatékony kezeléséhez viszont mindenképpen szükség van az előbb említett vizuális leckeszerkesztőre.

Javított TTS cache

A hangszintetizáló gyorsítótárának a hatékonyságát jelentősen javítani lehetne egy szöveg előfeldolgozóval, amely megpróbálja felépíteni a kapott kifejezést, korábban elhangzott kifejezésrészletekből.

Leckezáró párbeszéd

Egy lecke általában egy konkrét témához kötődő kifejezőkészletet tanít meg. A lecke zárásaként fel lehetne használni ezeket a kifejezéseket egy párbeszéd felépítésére is az alkalmazással, ahol a felhasználó élesben kipróbálhatná a szerzett tudását. Itt mindenféle instrukciók nélkül, a telefon csak kérdezne vagy mondana valamit, amire a felhasználónak reagálnia kell, majd erre a telefon válaszol, és szépen fel is épült egy természetes párbeszéd. Ha a tanuló sikeres, azzal egyben le is vizsgázott a leckéjéből. Ha valamire rosszul válaszol, akkor a hibásan visszaadott részeket a program újból átveszi.

Funkcionális fejlesztési lehetőségek***Emlékeztetők***

Manapság alapvetően bevett szokás a készülék értesítési sávjának a használata, értesítések küldésére a felhasználónak. Ezt egy nyelvoktató program esetén fel lehetne arra használni, hogy figyeljük mikor lépett be utoljára a felhasználó. Ezek után, ha azt tapasztaljuk, hogy túl rég óta nem gyakorolt, akkor erre emlékeztethetnénk, hiszen a nyelvtanulás egyik alapköve a rendszeresség. Persze ezek az emlékeztetők idővel zavarók lehetnek, szóval egy ilyen funkció implementálása esetén azt mindenképpen könnyen kikapcsolhatóvá kell tennünk.

Jutalmazás, képek, cikkek

Felmerült korábban annak a lehetősége, hogy a leckén belüli bizonyos mérföldkövek elérését a program érdekes történetekkel, vagy egyedi, a lecke anyagához kötődő dicséretekkel jutalmazná. Sőt, akár képeket, vagy később egyéb angol nyelvű cikkeket is beszúrhatnánk a felhasználó számára.

Üzleti fejlesztési lehetőségek

Érdekes kategóriába vennünk az üzleti megfontolású ötleteket és fejlesztési lehetőségeket.

In-app purchase alapú leckeletöltések

Az irodalomkutatás során láthattuk a Read4English [6] alkalmazás leckeletöltő menüjét, ahol alkalmazáson belül van lehetőségünk új leckéket letölteni ingyenesen, vagy akár pénzért is. Egy hasonló megoldással lehetne kezelni a mi alkalmazásunkban is az új leckék letöltését.

Eladható jutalmazás, reklámok, ajándékok

A korábban megtárgyalt jutalmazási rendszer felhasználható lehet üzleti célokra is. A felhasználó számára nemcsak érdekes, hanem irányított cikkeket is lehet adni jutalmazásként. Vegyük példának egy olyan leckét, ahol a tanuló az étteremben használatos kifejezésekkel ismerkedik meg. Ekkor a lecke végén nyugodtan ajánlhatunk neki éttermeket is a környékről, amelyek persze fizetett reklámok lennének. Persze mindezt a lecke hangulatába illő, intelligens módon kell tennünk.

Ábrajegyzék

1. ábra – A Tell Me More [1] nevű hangfelismerő algoritmust alkalmazó nyelvtanuló alkalmazás. Ez csak a kiejtés helyességére koncentrál, a kiejtett szöveg hangképének egy mintához való hasonlításával.	7
2. ábra – Az uTalk alkalmazás kategórianézete.	9
3. ábra - Advanced English Vocabulary pályaválasztója. Ebből is jól látszik az alkalmazás játékos jellege.	10
4. ábra - A Read4English alkalmazás "karaoke" jellegű felülete.	11
5. ábra – A példalecke irányított függőségi gráfja. Az ábra egyszerűsítésének érdekében egy kifejezés több pontként is megjelenhet.	20
6. ábra – A függőségek tárolásának szerkezeti rajza egy adatbázisban.	21
7. ábra - A program állapotdiagramja.	22
8. ábra – A program fő moduljai.	25
9. ábra – Képernyőkép az alkalmazás korai megjelenéséről. Itt a manapság népszerű szövegbuborékos csevegőstílust alkalmaztam. A zöld buborékok a felhasználó mondatai, a szürkék és a fehérek a telefon válaszai.	26
10. ábra – Képernyőkép az alkalmazás új designjáról. Ebben egy régi könyv formáját veszi fel, klasszikus párbeszédleírásokkal, és lapozási effektusokkal. A vörös mondatok a felhasználtól származnak, barnák és feketék a telefon válaszai.	27
11. ábra – Képernyőkép a trágár kifejezések szűrésének egy lehetséges, intelligens módjáról.	29
12. ábra – Az Android rendszer beépített adatforgalom mérőjének állásai a mérések után. A képernyőképek 10, és 35 felismert kifejezés után készültek. Látható hogy egy kifejezés átlagosan 1 KB adatforgalmat használ.	34

Irodalomjegyzék

- [1] J. Waltje, „Calico Journal - The Computer Assisted Language Instruction Consortium,” [Online]. Available: [https://calico.org/p-84-Tell%20Me%20More%20-%20German%20\(32001\).html](https://calico.org/p-84-Tell%20Me%20More%20-%20German%20(32001).html). [Hozzáférés dátuma: 5 11 2012].

- [2] „Siri,” Apple, [Online]. Available: <http://www.apple.com/ios/siri/>. [Hozzáférés dátuma: 5 11 2012].
- [3] „Google Now,” Google Inc., [Online]. Available: <http://www.google.com/landing/now/>. [Hozzáférés dátuma: 5 11 2012].
- [4] „uTalk for iPhone, iPod Touch, iPad and Android,” Eurotalk, [Online]. Available: <http://eurotalk.com/en/utalk/>. [Hozzáférés dátuma: 5 11 2012].
- [5] O. Manuel, „Google Play - Advanced English Vocabulary,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.oman.ln4kids>. [Hozzáférés dátuma: 5 11 2012].
- [6] „Read4English,” Brainglass, [Online]. Available: <http://www.brainglass.com/read4english/>. [Hozzáférés dátuma: 5 11 2012].
- [7] „The Michel Thomas Method: How it Works,” [Online]. Available: <http://www.michelthomas.com/how-it-works.php>. [Hozzáférés dátuma: 5 11 2012].
- [8] „The Pimsleur Method,” [Online]. Available: <http://www.pimsleur.com/The-Pimsleur-Method>. [Hozzáférés dátuma: 5 11 2012].
- [9] T. H. James Coady, *Second Language Vocabulary Acquisition: A Rationale for Pedagogy*, Cambridge: Cambridge University Press, 1997, p. 239.
- [10] „Android 4.1, Jelly Bean,” Google Inc., [Online]. Available: <http://www.android.com/about/jelly-bean/>. [Hozzáférés dátuma: 5 11 2012].
- [11] „Google Voice Search,” Google Inc., [Online]. Available: <http://www.google.com/mobile/voice-search/>. [Hozzáférés dátuma: 5 11 2012].
- [12] „Nuance - NDEV Mobile,” [Online]. Available: <http://dragonmobile.nuancemobiledeveloper.com/public/index.php?task=home>. [Hozzáférés dátuma: 5 11 2012].
- [13] „Google Play - Dragon Go!,” Nuance Communications, Inc, [Online]. Available: <https://play.google.com/store/apps/details?id=com.nuance.dragon>. [Hozzáférés dátuma: 5

11 2012].

- [14] „Swype Beta for Android,” Nuance Communications, Inc, [Online]. Available: <http://beta.swype.com/>. [Hozzáférés dátuma: 5 11 2012].
- [15] „CMU Sphinx - Open Source Toolkit For Speech Recognition,” Carnegie Mellon University, [Online]. Available: <http://cmusphinx.sourceforge.net/>. [Hozzáférés dátuma: 5 11 2012].
- [16] „SVOX,” SVOX Mobile Voices, [Online]. Available: <http://svoxmobilevoices.wordpress.com/>. [Hozzáférés dátuma: 5 11 2012].
- [17] „SVOX,” [Online]. Available: <http://www.svox.com>. [Hozzáférés dátuma: 5 11 2012].
- [18] „msdn: Model-View-Controller,” Microsoft Corporation, [Online]. Available: <http://msdn.microsoft.com/en-us/library/ff649643.aspx>. [Hozzáférés dátuma: 6 11 2012].