Szöveges adatbázis tervezése rendszerüzenet generátorhoz

NÉMETH GÉZA, OLASZY GÁBOR+, BŐHM TAMÁS
BME Távközlési és Médiainformatikai Tanszék, +MTA Nyelvtudományi Intézet,
{olaszy, nemeth, bohm}@tmit.bme.hu
UGRON ZOLTÁN
EMTE Sapientia, ugron.zoltan@gmail.com

Lektorált

Kulcsszavak: beszédválaszú rendszerek, korpusz-alapú beszédszintézis, szöveges adatbázisok, beszédadatbázisok

Beszédválaszú telefonos alkalmazások elsődleges kimenetei az előre felvett beszédüzenetek (prompt-ok) – a rendszer ezeknek a bejátszásával ismerteti a felhasználóval a hívott szolgáltatással kapcsolatos választási lehetőségeit (menüpontok), visszaigazolja műveleteit stb. A promptok szövegének alacsony entrópiája miatt valószínűsíthető, hogy az emberi beszédet megközelítő minőségben előállíthatóak egy erre a célra fejlesztett beszédszintetizátor segítségével. Ennek megvalósításával kiküszöbölhetők az új üzenetek hangfelvételi nehézségei. A sikeres szintézishez szükséges, hogy a promptgenerátor adatbázisa reprezentatív legyen a várható bemeneti adatokra, azaz az előállítandó promptokat minél kevesebb beszédelemből tudja összefűzni. Cikkünkben a fejlesztés alatt álló rendszer működési elvének ismertetése után a hangadatbázis elkészítéséhez szükséges, felolvasandó szöveges állomány (szövegkorpusz) tervezési módszerét tárgyaljuk, majd bemutatjuk, hogyan vizsgáltuk meg a korpusz szövegének reprezentativitását egy független szöveggyűjtemény felhasználásával.

1. Bevezetés

Napjainkra széles körben elterjedtek az interaktív beszédválaszú rendszerek. Ezeket elsősorban automatikus ügyfélszolgálatok és információs szolgáltatások megvalósítására használják.

A kimeneti funkció (beszédüzenet) megvalósítására kétféle megoldás ismert: rögzített hangfelvételek, mint rendszerüzenetek (promptok) vagy általános szöveg-beszéd átalakító által generált beszéd (text-to-speech, TTS). Az előbbi az összes lehetséges rendszerüzenet felvételét, tárolását, és megfelelő időben való lejátszását jelenti. Ebben az esetben csak előre rögzített beszédüzeneteket használhat a rendszer (például „Önnek új elektronikus levele érkezett"). Ez nem előnyös. További hátrány, hogy ezeknek a rögzített üzeneteknek a legkisebb változtatása esetén is új hangfelvételeket kell készíteni az eredeti bemondóval.

Sokkal rugalmasabb megoldás egy általános célú TTS alkalmazása, amely tetszőleges szöveget képes érthetően felolvasni. Így elég, ha az egyes rendszerüzenetek pontos szövege futási időben alakul ki (például: „Önnek új elektronikus levele érkezett Kovács Balázstól, melynek tárgya: holnapi találkozó"). A rendszer módosítása – például új funkciók bevezetése – esetén nem kell új hangfelvételeket készíteni. Ennek a megoldásnak is van hátrányos oldala; a gyengébb hangminőség. A ma, magyar nyelven elérhető TTS-ek által előállított beszéd jól érthető, emberi beszédre emlékeztető, de gépies hangzású. Az ilyen mesterséges beszéd még nem elfogadható rendszerüzenetként a felhasználók számára. Ez a magyarázata annak, hogy a hazánkban működő interaktív beszédválaszú rendszerek szinte kivétel nélkül előre rögzített promptokat használnak.

Megfigyelhető, hogy beszédválaszú rendszerben a bemondandó promptok szókincse jóval kisebb, mint általános szövegek esetén. A mondatok szerkezete, a fogalmazás módja is kevésbé változékony. Ennek oka, hogy a szövegek egy adott témához kapcsolódnak és egy adott (ügyfél-ügyintéző párbeszédre emlékeztető) stílust követnek. Fontos megjegyezni, hogy ettől még nem korlátozott az üzenetek témája. Bármikor, akár futási időben is, megjelenhetnek korábban nem látott szavak (például új szolgáltatások, termékek bevezetésekor). Így a szókészlet nem rögzített, de jelentősen eltolódik a témába eső szavak irányába. Ugyanez igaz a mondatszerkesztésre és a szófordulatokra is.

Az interaktív beszédválaszú rendszerek adott témáját kihasználva az alkalmazáshoz illesztett olyan TTS-t lehet létrehozni, amely sokkal jobb minőségű beszédet tud előállítani, mint egy általános célú beszédszintetizátor. A rögzített promptszövegek és az általános TTS előnyeit ötvözve jó minőségű beszédet előállító, de ugyanakkor rugalmas dialógusrendszerek építhetők.

A BME TMIT Beszédkutatási Laboratóriumában egy ilyen célra felhasználható promptgenerátor fejlesztése folyik, amely egy infokommunikációs szolgáltató beszédválaszú rendszereiben használható fel. egy a fenti célnak megfelelő promptgenerátor fejlesztése folyik, amely egy infokommunikációs szolgáltató beszédválaszú rendszereiben használható fel. A munka még nem zárult le, így cikkünkben a működési elv leírása után az egyik nagy súlyú kérdésre, a beszédadatbázis tervezésére koncentrálunk. Ismertetjük a beszédkorpusz tervezésének szempontjait, lépéseit és ellenőrzését.

2. Adott témájú szöveg-beszéd átalakító

Több adott témájú szöveg-beszéd átalakító készült már világnyelveken – például angolul [1] és németül [2]. Ezek beszédadatbázisa két részből tevődik össze. Az

adatbázis nagyobb része a tárgyterület jellemző szavait, kifejezéseit tartalmazza többféle szövegkörnyezetben. A másik része rövid hangsorépítő elemek (diádok, hangok vagy félhangok) teljes halmazát tartalmazza. Ez utóbbiak szolgálnak az olyan szövegrészek összeállítására, amelyek nem szerepelnek a célzott tematikájú adatbázisrészben – jellemzően a témán kívüli szavak, kifejezések.

A szintetikus beszéd előállítása ezekben a rendszerekben úgy történik, hogy az adatbázisból a megfelelő elemeket kiválasztják és összefűzik. Ezek a megoldások az általános TTS-ek körében elterjedt elemkiválasztási algoritmusokat alkalmazzák. Az elemkiválasztási algoritmusokról egy részletes ismertető és a kapcsolódó fogalmak (például költségfüggvények) definíciója [3]-ban olvasható,, illetve egy ilyen elven működő magyar nyelvű rendszer kerül ismertetésre [4]-ben. A bemeneti szövegből az említett rendszerek előállítják a szintézis célsorozatát – a szöveg fonetikus átiratát prozódiai információkkal. Meghatározzák az adatbáziselemek összes olyan sorozatát, ami a célsorozatot teljes egészében lefedi. Ezek közül a jelöltek közül azt választják, amelyik valamilyen költségfüggvényt minimalizál. A jelölt költsége a benne szereplő elemek célköltségeiből (mennyire jól reprezentálja a célsorozat megfelelő szakaszát) és az egymás utáni elemek összefűzési költségéből (mennyiben töri meg az akusztikai jel folytonosságát a két elem összefűzése) adódik össze.

Az előre ismert téma lehetővé teszi, hogy a korpusz nagy része témaspecifikus mondatokból álljon. Így nagy a valószínűsége, hogy a szintézis során hosszú, több szóból álló elemeket választ ki az algoritmus és így sokkal természetesebb hangzást ér el, mintha diádelemekből kellene építkezni. Black és Lenzo rendszere az adatbázisban egymással szomszédos (azaz az eredeti bemondásban is egymás mellett szereplő) elemek összefűzési költségét nullára állítja, így indirekt módon elősegítik a hosszú elemek kiválasztását [1].

Schweitzer és szerzőtársai ezzel szemben a jóval kevésbé számításigényes fonológiai struktúra-egyeztetés (phonological structure matching, PSM) módszert választották [2]. A PSM algoritmus először kideríti, hogy a teljes mondat vagy azok prozódiai egységei[1] teljes egészében megtalálhatóak-e az adatbázisban. Ha igen, akkor ezeket adja ki a kimeneten. Amelyik prozódiai egységet nem találta meg, annak szavait megkeresi és ezeket összefűzve állítja elő a kimenetet. Ha egy szó nem található vagy nem megfelelő környezetben és pozícióban található az adatbázisban, akkor azt szótagokból vagy végső esetben beszédhangokból fűzi össze.

Esetünkben a téma egy infokommunikációs szolgáltató telefonos beszédválaszú rendszereinek üzenetei. Első megközelítésként egy olyan rendszert szeretnénk építeni, amely a promptszövegek egy jól körülhatárolható részhalmazát emberi beszédet megközelítő minőségben képes szintetizálni. Ez a részhalmaz a „gomb" morfémát tartalmazó mondatok köre. A promptszövegek jelentős része tartalmazza ezt a betűsort – a menürendszerben való navigációt szolgáló, vagy a mobiltelefon beállításokat ismertető üzenetekben. Bár a „gomb" mondatok teszik ki a jelenlegi beszédválaszú rendszerek üzeneteinek jelentős részét, ezek entrópiája jóval alacsonyabb a többi mondatnál (amelyek általában hosszabb ismertetők részei, így témájuk és szókincsük jóval változatosabb). Ilyen megfontolások miatt valószínűsíthető, hogy a „gomb"-os mondatokra jó minőségű adott témájú TTS fejleszthető. Ennek a rendszernek a fejlesztése során szerzett tapasztalatok és az elkészült komponensek később felhasználhatók lesznek egy tetszőleges promptszöveget generáló rendszer kidolgozásához.

A „gomb"-os mondatok alacsony változékonyságát figyelembe véve azt tűztük ki célul, hogy a rendszer az esetek többségében képes legyen a kimenetét előre felvett prozódiai egységekből összefűzni. Ha egy szükséges prozódiai egység nem található meg az adatbázisban, akkor azt szavakból fűzzük össze – hasonlóan a PSM megközelítéshez. Ha ez nem sikerül, akkor az adott ponton a rendszer még rövidebb elemekből állítja elő a beszédet. Az elemkiválasztási algoritmus és a szintézis folyamata terveink szerint nagyon hasonló lesz az időjárásjelentés témakörére kidolgozott felolvasóban alkalmazotthoz [4]. A felvételek címkézésére is az ott kidolgozott módszereket fogjuk felhasználni.

3. Korpusztervezés

A fejlesztés első lépése a szöveges adatbázis (szövegkorpusz) megtervezése. Ennek a felolvasott változata a folyamatos beszédet tartalmazó beszédadatbázis, melyből az elemkiválasztó algoritmus tetszőleges hosszúságú (prozódiai egység, szó, diád stb.) elemeket kivághat és beilleszthet a kimeneti jelbe. Az alábbiakban az ehhez kidolgozott módszereket ismertetjük.

A szöveges adatbázis tervezése során két fontos szempontot kell figyelembe venni:

1. Megfelelően kell lefedje a lehetséges, adott témájú szövegeket. Tehát a témához tartozó mondatok szintetizálása során minél hosszabb elemek teljes egészében legyenek benne a szövegkorpusz felolvasása és címkézése után létrehozott beszédadatbázisban.
2. Az általánosság biztosítására legyen lehetőség a témától független szövegek – esetleg rövidebb elemekből történő – összefűzésére is.

Az ismert megoldások [1,2] külön kezelik a két követelményt: felvesznek a témához illeszkedő mondatokat és egy külön mondathalmazzal biztosítják a teljes diádfedést. Úgy gondoljuk, hogy nem feltétlenül szükséges ez a szétválasztás. Egy nagyméretű, adott témájú beszédkorpusz valószínűleg a diádok túlnyomó

1 Prozódiai egységnek nevezzük a beszédnek azt a szakaszát, amely a dallammenet szempontjából egy egységet alkot. A prozódiai egységek gyakran egybeesnek a tagmondatokkal.

többségét tartalmazza. A felolvasási listát ki lehet egészíteni a maradék diádokat tartalmazó mondatokkal.

Ahhoz, hogy minél több, a területre jellemző mondatot tudjunk felvenni, egy infokommunikációs szolgáltató nagy mennyiségű promptszöveget bocsátott a rendelkezésünkre. Ez a szöveghalmaz a 2000. január és 2005. június között felvett és a szolgáltató beszédválaszú rendszereibe beépített promptokat tartalmazta. Ebből a tanítóhalmazból alakítottuk ki a felolvasandó mondathalmaz első változatát.

3.1. Szövegtisztítás és -normalizálás

A további feldolgozás azt igényelte, hogy a rendelkezésünkre bocsátott prompt szöveghalmazból kinyerjük a tényleges promptszövegeket és azokat egységes formátumra hozzuk. Ezt automatikus és félautomatikus módszerekkel végeztük el. Először kiszűrtük azokat a szövegrészeket, amelyek nem részei a promptok szövegének, például a promptok azonosítószáma vagy státusza. Az idegen nyelvű promptokat is eltávolítottuk, de az idegen szavakat tartalmazó magyar üzeneteket meghagytuk, mert ezeket kezelnie kell a rendszernek. Töröltük továbbá a zárójeleket tartalmazó promptokat, mert ezek különleges kezelést igényelnek a felolvasáskor.

A promptlistát egy táblázatként kaptuk meg, cellánként egy prompttal (ami több mondatból is állhat). Ha a cellák szövegének végén nem volt pont, azt pótoltuk. A táblázatot szövegfájllá alakítottuk és automatikusan mondatokra tördeltük, így minden sorba pontosan egy mondat került. Végül az egészet kis betűssé alakítottuk, mert a későbbiekben a kisbetű-nagybetű különbségeket szeretnénk figyelmen kívül hagyni. Erre azért van szükség, mert nyilvánvalóan a „KÜLDÉS", „Küldés" és „küldés" szavak felolvasása teljesen megegyezik.

Így előállt a teljes promptgyűjtemény normalizált formában, amely több, mint 39 ezer mondatot tartalmazott. Ebből kiválogattuk a „gomb" karakterláncot tartalmazó mondatokat és a továbbiakban csak ezekkel dolgoztunk. Ez 4189 mondatot jelent.

3.2. Kategorizálás

A „gomb"-ot tartalmazó mondatok halmazát tovább osztottuk, hogy a különböző, tipikus mondatszerkezetek gyakoriságát és tulajdonságait megvizsgálhassuk. A 4189 mondatot (1596 egyedi) automatikusan az alábbi öt kategória egyikébe soroltuk (*dőlt betűvel* egy-egy példát is megadunk):[2]

– „gomb" szóval végződő mondatok:
  Magánszemélyek, 3-as gomb.
– „nyomja meg a(z) X gombot" kifejezéssel végződő mondatok, ahol X helyén egy szónak megfelelő karaktersorozat áll:
  A kód beírása után nyomja meg a # gombot.
– „nyomja meg a(z) X gombot" kifejezést tartalmazó, de nem azzal végződő mondatok:
  Nyomja meg az OK gombot, és válassza a módosítás opciót.
– „nyomja meg a(z)" kifejezést tartalmazó, de nem az előbbi két kategóriába tartozó mondatok:
  Nyomja meg a boríték jelű gombot hosszan.
– a fenti kategóriák egyikébe se tartozó mondatok:
  *Pluszjelet a * gomb kétszeri megnyomásával írhat.*

Az egyes kategóriákba sorolt mondatok száma az *1. táblázatban* látható. Az „Összes" oszlopban olvasható a megfelelő kategóriájú mondatok összes előfordulásának száma, míg az „Egyedi" oszlopban a legalább egy karakterben különböző mondatok száma.

Kategória	Összes	Egyedi	Lefedő
„gomb végű"	3236	1153	1008
„nyomja meg a(z) X gombot" végű	148	88	81
„nyomja meg a(z) X gombot"	91	36	36
„nyomja meg a(z)"	125	75	67
egyéb	589	244	239
Összesen	**4189**	**1596**	**1431**

1. táblázat
A „gomb" szót tartalmazó és azok teljes lefedéséhez szükséges mondatok száma kategóriánként

Ezután a mondatokat automatikusan prozódiai egységekre tördeltük, így minden prozódiai egység külön sorba került. A tördelés a prozódiai egységek határjelzői (például mondatvégi írásjelek, vessző vagy kettőspont) alapján történt. Mindegyik egységhez eltároltuk a mondatbeli pozícióját is: kezdő/egyedüli, belső vagy záró. Erre azért volt szükség, mert a szintézis során csak a megfelelő pozícióból kivágott prozódiai egységeket célszerű felhasználni a kielégítő prozódia megvalósításának biztosítására. Például nem lehet egy mondatzáró prozódiai egységet egy mondat első egységeként beilleszteni, mert az egész dallammenet természetellenes lesz. A pozícióból eredő különbségeket jól illusztrálja a szóhasználatbeli eltérés – ahogy az a *2. táblázatból* látszik, a leggyakoribb szavak listája jelentős eltérést mutat.

2. táblázat
A promptszövegekben a 10 leggyakrabban előforduló szó a prozódiai egység mondatbeli pozíciója szerint

	Mondatkezdő/egyedüli prozódiai egységekben	Belső prozódiai egységekben	Mondatzáró prozódiai egységekben
1.	a	a	a
2.	az	ft	gomb
3.	és	és	és
4.	gomb	az	az
5.	szolgáltatás	következő	ft
6.	domino	wap	vagy
7.	sms	válasszon	hogy
8.	t	sms	t
9.	kérjük	hogy	sms
10.	ft	t	is

Látható, hogy legalább három prozódiai egység pozíció szerinti megkülönböztetése indokolt. A beszédszintézis területén szerzett tapasztalataink azt mutatják, hogy ennél több pozíció használata nem szükséges.

2 A könnyebb olvashatóság érdekében itt a mondatok a kisbetűssé alakítás előtti formában szerepelnek.

Utolsó lépésként minden kategória prozódiai egységeinek listájából kiszűrtük az ismétlődéseket (a kizárólag a pozícióban eltérőeket különbözőnek tekintve).

Eltárolhattuk volna még a mondatok modalitását is (kijelentő, kérdő, felszólító, felkiáltó vagy óhajtó), erre azonban a promptok témája esetén nincs szükség. Minden promptszövegben előforduló mondat kijelentő vagy felszólító, melyek dallammenete nagyrészt megegyezik.

3.3. A teljes fedést biztosító mondatok kiválasztása

Mind az öt kategóriához adott a szövegkorpusz mondatainak listája és azok felbontása prozódiai egységekre. A cél egy olyan minimális mondathalmaz összeállítása, amely tartalmazza az összes prozódiai egységet legalább egyszer. Ennek a halmaz-fedési problémának egy jól ismert közelítő megoldása a mohó algoritmus [5]. Minden lépésben egy mondatot adunk hozzá a fedőhalmazhoz. Az első kiválasztott mondat az, amelyik a legtöbb új egységet tartalmazza; utána minden lépésben azt a mondatot adjuk hozzá, amelyik a legtöbb, még lefedetlen egységet tartalmazza; ezt addig ismételjük, amíg van lefedetlen egység. A mohó algoritmus többféle továbbfejlesztése ismert [6], de ezek ennél a problémánál nem jelentenek előnyt.

A mohó algoritmus futtatásával megkaptuk a felolvasandó mondatok listájának első változatát. Kategóriánként a kiválasztott mondatok számát az 1. táblázat foglalta össze. Az eredeti mondatszámhoz képest kis mértékű, körülbelül 10%-os csökkenést értünk el.

A módszerrel kapcsolatban két dolgot érdemes kiemelni. Az egyik az, hogy végig a szövegek írott formájával dolgoztunk, nem a fonetikus átiratukkal. Ez bizonyos mértékű hibát okoz az eredményekben – ha egy prozódiai egységnek két, különbözőképpen leírt formája szerepelt a szövegekben, akkor az az egység duplán szerepel a felolvasási listán. Ezek száma valószínűleg kevés. Az előállított beszéd minőségét ezek a „hibák" nem rontják, sőt, egyes esetekben javíthatják (ugyanabból az elemből több alternatíva áll rendelkezésre, így az elemkiválasztó algoritmus az adott célsorozathoz leginkább illeszkedőt tudja kiválasztani).

Ha azonban betű-beszédhang átalakítást alkalmaztunk volna a mohó algoritmus futtatása előtt, az számos problémát vetett volna fel a betű-beszédhang algoritmusok tökéletlensége miatt. Schweitzer és szerzőtársai 170 ezer német mondatból 70 ezerben találtak átírási hibát [2]. Magyar nyelv esetén ez kisebb problémát okoz, mert maga a betű-beszédhang átalakítás egyszerűbb. Viszont elengedhetetlen az ezt megelőző betű-betű átalakítás, amely többek között a rövidítések, számok kifejtését, az idegen és rendhagyó írásmódú szavak átírását jelenti. Ez az algoritmus számos esetben téved és ezeket a tévedéseket nehéz automatikusan felderíteni.

A másik megjegyzés a gyakoriságokkal kapcsolatos. Nyilván érdemes lenne figyelembe venni az egyes prozódiai egységek előfordulásának gyakoriságát is. Viszont az, hogy a promptok szövegeiben hányszor fordul elő egy-egy egység, független attól, hogy milyen sűrűn játssza le azt az egységet a rendszer. Tehát a rendelkezésre álló adatok alapján számolt gyakoriságok valószínűleg félrevezetőek. Van Santen és Buchsbaum szerint még akkor sem érdemes a gyakorisági információt felhasználni, ha azok elérhetőek [5]. Ennek oka, hogy ezek a gyakoriságok csupán egy időben változó eloszlás mintái – jó példa erre a „WAP" és a „telex" szavak, melyek előfordulásainak száma nagyságrendileg eltérő lehet egy 1995-ös és egy 2005-ös promptgyűjtemény között.

4. A szövegkorpusz ellenőrzése

Annak érdekében, hogy a koncepció helyességét ellenőrizzük, egy független teszthalmaz fedését is kiszámítottuk. A teszthalmaz egy infokommunikációs szolgáltató két telefonos beszédválaszú rendszerében 2005. novemberben használt összes prompt szöveges formája. A tanító- és a teszthalmaz méreteit a 3. táblázat foglalja össze.

	Tanítóhalmaz	Teszthalmaz
Mondatok (ismétlődésekkel)	39 247	5 666
Szavak	530 526	61 225
Szóalakok	8 914	5 621
Prozódiai egységek	82 642	11 968
Egyedi prozódiai egységek	17 849	7 286

3. táblázat A tanító- és a teszthalmaz méretei

Kiszámítottuk, hogy a teszthalmaz prozódiai egységeinek hány százalékát fedik le a felolvasólista mondatai. Akkor tekintettünk egy egységet lefedettnek, ha a felolvasólistán van egy olyan egység, amelyik ugyanazt a karaktersorozatot tartalmazza és a mondatban ugyanabban a pozícióban van. Az eredmények a prozódiai egység kategóriák és pozíciók szerint a *4. és 5. táblázatban* láthatóak.

A teljes fedési arány nagyon magas, 76%. Ez azt jelenti, hogy ha a promptgenerátort a felolvasási lista első változata alapján valósítanánk meg és a teszthalmaz teljes szövegét szintetizálnánk a rendszerrel, akkor négyből három prozódiai egység emberi beszédet megközelítő minőségben szólna.

4. és 5. táblázat
A teszthalmaz összes egyedi prozódiai egységeinek és a lefedetlenek száma kategóriák és pozició szerint

Kategória	Összes	Lefedetlen
„gomb végű"	839	206 (25%)
„nyomja meg a(z) X gombot" végű	63	17 (27%)
„nyomja meg a(z) X gombot"	50	14 (28%)
„nyomja meg a(z)"	35	4 (11%)
egyéb	235	51 (22%)
Összesen	1222	294 (24%)

Pozíció	Összes	Lefedetlen
Mondatkezdő/egyedüli	785	132 (17%)
Belső	288	119 (41%)
Mondatzáró	149	43 (29%)
Összesen	1222	294 (24%)

Az egyedi szóalakok fedése 95%. Előzetes feltételezésünk az volt, hogy ha magas fedési arányt tudunk elérni a prozódiai egységek szintjén, akkor az implicit módon magas szószintű fedési arányt is jelent. Ez a feltételezés bebizonyosodott. Bár a szószintű fedésnél csak a szavak karaktereinek egyezését néztük (így például a pozíciót, környezetet, hangsúlyosságot figyelmen kívül hagyva), valószínűsíthető, hogy azokban az esetekben, amikor a cél prozódiai egység nem található meg az adatbázisban, a rendszer legtöbbször képes azt szavakból összefűzni. Ezek alapján feltételezzük, hogy a promptgenerátorral a jelenleg használatos diád- és triád-alapú rendszerek beszédének természetességét messze felül tudjuk múlni.

Érdekes megfigyelés, hogy a lefedetlen szóalakok száma (310) nagyjából megegyezik a lefedetlen prozódiai egységek számával. Lehetséges, hogy a legtöbb lefedetlen prozódiai egység mindössze egyetlen szóban tér el a tanítóhalmazban hozzá legközelebb levő prozódiai egységtől és ez az egy szó egyáltalán nem szerepelt a tanítóhalmazban.

5. Összefoglalás és kitekintés

Cikkünkben bemutattuk egy készülő promptgenerátor beszédkorpuszának tervezési folyamatát. A rendelkezésre álló promptszövegeket normalizáltuk, kategóriákra bontottuk, majd a mondatokból mohó algoritmussal egy (prozódiai egységek szempontjából) teljes fedést biztosító részhalmazt választottunk ki. Az összeállított mondathalmaz egy független teszthalmaz prozódiai egységeinek 76%-át lefedte. A szószintű fedés 95% volt. Ezek az adatok azt mutatják, hogy elképzelésünk működőképes.

A végleges felolvasólista a 3. szakaszban ismertetettnél nagyobb lesz, mert azon felül az alábbiakat is tartalmazni fogja:

- Olyan mondatok, amelyek hozzáadásával elérhető, hogy a teszthalmazra is 100%-os legyen a prozódiai egységek fedése. Ez körülbelül 300 új mondat hozzáadását jelenti.
- Az esetlegesen hiányzó diádokat tartalmazó bemondások.
- A számok megfelelő kezeléséhez a teljes számelemkészlet [7].
- A betűszavak jó minőségű szintéziséhez az összes betűelem (például „emm", „zé" és „iksz").

A bemutatott módszernek több ismert hiányossága van. A mohó algoritmus a hosszú mondatokat preferálja a rövidek helyett, pedig az utóbbiakat sokkal könnyebb helyesen felolvasni a korpusz felvételekor. Ez elsősorban a keresés elején igaz, amikor még szinte minden prozódiai egység lefedetlen, így egy hosszabb mondat több újonnan lefedett egységet eredményez. Az azonos számú lefedetlen egységet tartalmazó mondatok közül pedig a listában először szereplőt választja. Az utóbbi probléma azonban orvosolható: azonos számú új egységet tartalmazó mondatok közül mindig a rövidebbet kell választani.

Jelenleg a mohó algoritmust mind az öt kategóriára külön futtattuk, hogy jobban megismerjük azok eloszlását. A végleges felolvasási lista mérete azonban kissé csökkenthető, ha egyszerre futtatjuk az összes kategóriára az algoritmust, ugyanis az egyes kategóriák között lehetnek átfedések.

Az egyes prozódiai egységek lejátszásának gyakorisága alapján tovább javítható a korpusz összetétele: például a leggyakoribb egységeket többször is felvehetjük, míg a nagyon ritkákat kihagyhatjuk. Jelenleg ilyen statisztikai adatok nem állnak rendelkezésünkre, azonban a rendszer béta változatának beindításától kezdve a naplófájlokból kinyerhetőek.

A továbblépés szempontjából fontos kérdés, hogy az összes prompt mondat (nem csak a „gomb"-ot tartalmazók) teljes fedéséhez hány további mondat felvétele szükséges. Egyenlőre nem egyértelmű, hogy a prozódiai egység szintű megközelítés ebben az általánosabb esetben milyen eredményre vezet.

Köszönetnyilvánítás

Ezt a kutatást az NKFP 2. program (szerződésszám: 2/034/2004) támogatta.

Irodalom

[1] Black, A. W., Lenzo, K. A.,
Limited domain synthesis, Proc. of ICSLP 2000.
[2] Schweitzer, A., Braunschweiler, N., Klankert, T., Möbius, B., Sauberlich, B.:
Restricted unlimited domain synthesis,
Proc. of Eurospeech 2003, pp.1321–1324.
[3] Möbius, B.,
Corpus-based speech synthesis: Methods & challenges,
Arbeitspapiere des Instituts für
Maschinelle Sprachverarbeitung (Univ. of Stuttgart),
Vol. 6, No.4, 2000, pp.87–116.
[4] Fék, M., Pesti, P., Németh, G., Zainkó Cs.:
Generációváltás a beszédszintézisben,
Híradástechnika, jelen számban
[5] van Santen, J. P. H., Buchsbaum, A. L.,
Methods for optimal text selection,
Proc. of Eurospeech 1997, Vol. 2., pp.553–556.
[6] Bozkurt, B., Ozturk, O., Dutoit, T.,
Text design for TTS speech corpus building using a modified greedy selection,
Proc. of Eurospeech 2003, pp.277–280.
[7] Olaszy G., Németh G.:
IVR for banking and residential telephone subscribers using stored messages combined with a new number-to-speech synthesis method,
In Gardner-Bonneau, D. (ed.): Human Factors and Interactive Voice Response Systems, Kluwer, 1999.
pp.237–255.