Diktálórendszer pontosságának és hatékonyságának vizsgálata a keresési téren alkalmazott vágási technikák függvényében

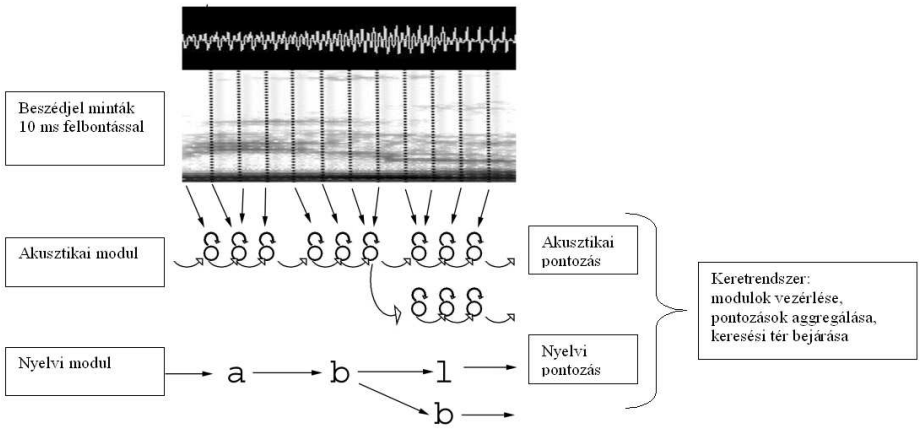
Bánhalmi András, Paczolay Dénes, Tóth László

MTA-SZTE Mesterséges Intelligencia Kutatócsoport
6720 Szeged, Aradi vértanúk tere 1.
{banhalmi,pdenes,tothl}@inf.u-szeged.hu

Kivonat Folyamatos beszéd felismerése esetén a beszédjelhez illeszthető szósorozatok száma exponenciálisan nő a felvétel hosszával. Ezért a diktálórendszerek hatékonysága szempontjából kulcsszerepe van a különböző, a keresési teret redukáló vágási technikáknak, illetve kiértékelést gyorsító trükköknek. A keresési tér vágásával elért sebességnövekedés könnyen a felismerési pontosság rovására mehet, ezért a módszerek paramétereinek beállításakor meg kell találni a megfelelő egyensúlyt a hatékonyság és a pontosság között. Cikkünkben bemutatjuk, hogy az általunk fejlesztett felismerő hogyan reprezentálja a keresési teret, az azt nagyban meghatározó nyelvi komponenst, továbbá hogy maga a keresés hogyan történik. Ismertetjük, hogy a keresés során milyen vágási technikákat alkalamazunk, majd konkrét felismerési teszteken keresztül megvizsgáljuk, hogy különböző paraméterértékek mellett ezek hogyan befolyásolják a futási időt és a felismerési pontosságot.

1. Motiváció

Folyamatos diktálórendszerek konkrét technikai megvalósításával kapcsolatosan viszonylag kevés szakirodalom hozzáférhető (néhány elterjedtebb módszer leírása itt megtalálható: [1], [2], [3]). Ennek talán az lehet az oka, hogy a keresési teret bejáró algoritmusokon és különféle hatékony számolási technikákon nagyban múlik az, hogy egy diktálórendszer mennyire lesz gyors és mennyire lesz pontos. Így ezek a technikák piaci szempontból nagy jelentőséggel bírnak. Ebben a cikkünkben egy részletes leírást adunk a saját fejlesztésű diktálórend-szerünk felépítéséről, az egyes modulok processzorigényéről, különböző gyorsítási lehetőségekről, és a keresési tér általunk alkalmazott vágási módszereiről. Elemezzük, hogy az egyes vágási technikák hogyan befolyásolják a felismerési pontosságot és a processzorigényt.



1. ábra. Diktálórendszer moduláris szerkezete

2. A diktálórendszer felépítése

Egy diktálórendszer alapvető feladata abban áll, hogy egy mikrofonba bemondott beszédjelet szöveges információvá alakítson át. Ehhez – a megvalósítás szintjén – két, egymástól eltérő szerepű eszköz áll rendelkezésére. Az egyik a nyelvi modul, aminek a feladata az, hogy egy szósorozathoz pontértéket rendeljen aszerint, hogy az adott szósorozat mennyire valószínű. Ennek a feladatnak a megoldására többféle modellt is javasoltak már. Egy nyelvi modell annál jobb, minél inkább „kiszűri" azokat a szósorozatokat, amik csak nem tipikus diktálás mellett fordulhatnak elő, de nem pontozza le azokat a szósorozatokat, amik ha nem is gyakran, de előfordulhatnak a diktálás során. A másik fontos eleme egy diktálórendszernek az akusztikai modul, aminek a feladata a különböző beszédhangok modellezése valamilyen gépi tanuló algoritmus segítségével. Diktáláskor az akusztikai modul értékeli azt, hogy a bediktált jel egy darabja mennyire tartozik bele egy beszédhang-osztályba. Egy valós idejű diktálórendszer az említett két modul pontozását kombinálva rangsorolja a lehetséges beszédhangsorozatokat, ezekből egy keresési teret építve folyamatosan a diktálás ideje alatt. Hogy ez a tér ne növekedjen exponenciálisan, és így a rendszer hatékony tudjon maradni, különböző heurisztikus vágási technikákat kell alkalmazni. A rangsorolást, a keresési tér felépítését és a vágásokat a nyelvi és akusztikai modulokra épülő keretrendszer valósítja meg (ld. 1. ábra). Ebben a fejezetben részletesen írunk a saját rendszerünkbe beépített modulokról.

2.1. Nyelvi modul

Egy nyelvi modul alapfeladata egy olyan függvénynek a megtanulása és hatékony kiértékelése, ami egy szósorozathoz megad egy pontértéket. A nyelvi modul a mi

implementációnkban a pontértékek hozzárendelése mellett az adott területhez (domain) tartozó lehetséges szósorozatok „bejárására” (generálására) is képes, ezzel támogatva a keresést.

Ezen felül - a mi felfogásunkban - a nyelvi modul nem lehetséges szósorozatokat, hanem lehetséges beszédhangsorozatokat generál (ugyanazon szósorozathoz többféle beszédhangsorozat is rendelhető). A mi megvalósításunkban a nyelvi modulnak meg kell adnia azokat a beszédhangokat, amelyek a modul egy adott állapotában következhetnek. Emellett a modul megadja még a lehetséges folytatások bizonyos jellemzőit is, amik arról adnak információt, hogy egy szó végére értünk-e, folytatható-e tovább a nyelvi kiértékelés, valamint megadja az egyes lehetőségek nyelvi pontozását (valószínűségét).

Egy példán szemléltetve: a nyelvi modul állapota legyen ("klinikai adatok", "k l i n i k a j i j a d a t o k _ v á l"), aminek a jelentése az, hogy az eddig bemondott teljes szavak a "klinikai adatok" voltak, a szavak között hasonulást tételeztünk fel a beszédhangsorozat átiratában, és a harmadik, még ismeretlen szóból a "v á l" beszédhangsorozatig jutottunk el. A nyelvi modul ebben az állapotában visszaadja a lehetséges folytatásokat (pl. "t", "l", "o", "a", stb.), a hozzájuk tartozó pontértéket, valamint szóvégre és nyelvtanvégre hamisat ad vissza. Egy ilyen függvény többféleképpen is megvalósítható, és sokféle nyelvi információt felhasználhat a pontérték megadásához. A mi modulunk három nyelvi modellel rendelkezik jelenleg. Ezek a szó N-gram és ennek simított változatai [4], a környezetfüggetlen nyelvtan, és az MSD-kód alapú csoport N-gram (erről részletesebben [5]). Mivel szó N-gram használatakor igen nehézkes megadni számokat, dátumokat, neveket, ezért a szó N-gram megvalósításunkban lehetőség van szavak helyett szavak egy-egy halmazát megadni, és ezek a halmazok leírhatók egyszerű szó-listával, illetve szabályokkal is. A nyelvi modul a következő lehetőségeket tartalmazza:

- Egy szótár megadása kötelező.
- Szó N-gramok adhatók meg.
- Hasonulási szabályok adhatók meg.
- A szótár szavai csoportokba sorolhatók.
- A csoportokra környezetfüggetlen nyelvtan adható meg.
- A csoportokra N-gram adható meg.

A konkrét megvalósítás oldaláról nézve, attól függően, hogy az előbbi lehetőségek közül melyeket használjuk nyelvi modellezésre, különböző struktúrákat épít fel a nyelvi modul. Legegyszerűbb esetben egy prefix-fa épül fel, amelynek a leveleit visszakötjük a fa elejéhez. Hasonulási szabályok használata esetén már egy ennél bonyolultabb gráfot állítunk elő, amelyben az egyes hasonulási lehetőségekhez tartozó ágak később egyesülnek. Környezetfüggetlen nyelvtan használata esetében pedig egy összetett, prefix-fákból álló gráfot építünk fel, ami utána minimalizálásra kerül. A minimalizálásnál az a célfüggvény, hogy azonos szósorozat illetve hozzá tartozó beszédhangsorozat ne jelenjen meg a gráfban egynél több

ágon. Ez alól egy kivétel van, ami akkor jelentkezhet, ha egy szó több csoportba is tartozhat. Ennek a felépítésnek az alapvető oka, illetve célja, hogy az aktuális hipotézislistában (ami beszédhangsorozatokból áll) kétszer ne jelenhessen meg ugyanaz a beszédhangsorozat, azaz a nyelvi modul ne generálhasson azonos hipotéziseket (mert ezek elvehetik a helyet a többi hipotézistől).

A nyelvi modult úgy valósítottuk meg, hogy amilyen hamar csak lehet, megadja a szó- valamint csoportsorozathoz tartozó pontértéket. Ez azt jelenti, hogy amikor már egyértelmű, hogy melyik szó (illetve csoport) tartozik a beszédhangsorozathoz (még nem feltétlenül értük el a szó végét), akkor a nyelvi modul már megadja a megfelelő pontértéket. Ennek az ún. előrehozott kiértékelési módszernek továbbfejlesztéseként olyan technikát is alkalmazunk, ami már az egyértelművé válás előtt egy (az ágakhoz tartozó maximummal) becsült értéket, majd később korrekciós értékeket ad meg pontozáskor. Annak a fontosságát, hogy a nyelvi modul minél hamarabb megfelelően értékelni tudja a beszédhangsorozatokat, egy korábbi cikkünkben elemeztük [5].

2.2. Akusztikai modul

A diktálórendszerünkbe az általánosan elterjedt Rejtett Markov Model (HMM) alapú akusztikus modult építettük be. Minden beszédhangra egy-egy balról-jobbra struktúrájú HMM-et tanítunk.

Diktáláskor egy-egy beszédhangsorozatnak egy-egy folyamatosan felépülő HMM lánc felel meg. Beszédfelismerés során a nyelvi modul lekérdezésével bővítjük a láncokat. Ezzel párhuzamosan az akusztikai modul feladata az, hogy a feldolgozásban soron következő beszédjel-szelet alapján az aktuális (a láncok végén szereplő) HMM-ek valószínűségi adatait megfelelően módosítsa.

A HMM-ek minden állapotához hozzá van rendelve egy valószínűségi eloszlás (GMM, Gaussian Mixture Modell, Gauss eloszlások súlyozott összege). Alapesetben az akusztikus modell a HMM-ek minden elérhető állapotára kiszámolja a következő értéket:

$$p(x) = \sum_{i=1}^{M} w_i \frac{1}{2\pi \left|\Sigma_i\right|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}, \quad (\sum_{i=1}^{M} w_i = 1),$$

ahol M a Gauss-komponensek száma, w_i a komponensek súlya, Σ_i és μ_i a komponensekhez tartozó kovarianciamátrix és középérték-vektor, és x a jellemzővektor. A gépi számábrázolás korlátozottsága és a számítások egyszerűsítése érdekében az előző értéknek a logaritmusát szokás kiszámítani. Ehhez a számításhoz a logaritmikus aritmetikából a következő, összeadásra vonatkozó képletet használhatjuk, így végig logaritmikus aritmetikában maradva elkerülhetjük a lebegőpontos alulcsordulást:

$$\log(a+b) = \log(e^{\log a} + e^{\log b}) = \log(a) + \log(1 + e^{\log(b) - \log(a)})$$

További egyszerűsítésként az összegnek a maximummal való közelítése ajánlott (a két érték között nincsen lényeges eltérés a gyakorlatban):

$$p(x) \approx \max_i(\log(w_i \frac{1}{2\pi|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}))$$
$$= \max_i(\log(\frac{w_i}{2\pi|\Sigma_i|^{1/2}}) - \frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i))$$

A kifejezés első tagja egy x-től független, előre kiszámítható konstans. A második taggal kapcsolatosan a beszédfelismerésben általánosan elterjedt gyakorlat (a számítások csökkentése érdekében), hogy a négyzetes kovariancia mátrixot diagonálisnak tételezzük fel. Így a következő számítást kell csak elvégeznünk:

$$\frac{1}{2}\sum_j \frac{(x_j - (\mu_i)_j)^2}{(\Sigma_i)_{jj}}$$

Ezt a számítást nem kell teljes egészében minden - az adott állapothoz tartozó - Gauss-komponensre elvégezni: ha az összegzés során az érték az addigi maximális érték alá csökkent, akkor a számítást nem fejezzük be. Ez a módszer még tovább gyorsítható úgy, hogy a Gauss-komponensek kiszámításának sorrendjét előre lerendezzük azok súlya szerinti csökkenő sorrendbe.

Mindezek mellett egy akusztikai alapú büntető, illetve szűrő eljárás használatát is javasoljuk, amellyel még nem találkoztunk a szakirodalomban. Az ún. egyosztályos statisztikai modellekben egy – a pozitív tanuló példákon számított – ***konfidenciaérték*** alapján sorolják be a tesztpéldákat pozitív példának (osztályba tartozónak), illetve negatív példának (nem az osztályba tartozónak). Rejtett Markov Modell alapú beszédfelismerő modellek esetében - a modellek tanításának kiegészítéseként - mi is meghatározunk minden HMM minden Gauss komponenséhez egy-egy konfidencia értéket. Ha a beszédfelismerés során az akusztikus modell által számított pontérték a konfidenciaértéknél kisebb lesz, akkor a számítást befejezzük, és egy megfelelően kicsi ponttal értékeljük a beszédjelet az adott Gauss-komponensre nézve. A konfidenciaértéket a pozitív tanulóhalmaz legkisebb pontértékéhez viszonyítva adjuk meg, ez nálunk egy 1,5-szörös szorzással adódik (logaritmikus aritmetikában számolva).

3. Keretrendszer a keresési tér felépítésére

Leegyszerűsítve, a keretrendszer feladata abban határozható meg, hogy fel kell építenie a hipotéziseknek egy olyan terét, amelyet a nyelvi modul megenged, és az egyes hipotézisekhez pontértéket kell rendelnie a nyelvi és az akusztikai pontozás alapján. A cél a legnagyobb ponttal rendelkező hipotézis(ek) megtalálása. Saját megvalósításunkban - alapjaiban - egy Viterbi N-best típusú keresést használunk [6], ami azt jelenti, hogy a keresési térnek mindig csak a legjobb N hipotézisét terjesztjük ki. Ennél az egyszerű módszernél azonban több ponton bővül a mi megközelítésünk, amit ebben a fejezetben részletesen bemutatunk.

3.1. A hipotézis fogalma

Egy hipotézist a következő adatok határoznak meg: (időpont, csoportsorozat, szósorozat, beszédhangsorozat, aktuális HMM azonosító, az aktuális HMM állapotaihoz tartozó valószínűségi értékek). Bevezetjük a következő jelölést:

$$H(t, [C_1 \dots C_k], [W_1 \dots W_k], [Ph_1 \dots Ph_n], \Theta_{Ph_n}, [p_0, p_1, p_2, p_3, p_4])$$

Itt t az időpontot, $[C_1 \dots C_k]$ a csoportsorozatot, $[W_1 \dots W_k]$ a szósorozatot, $[Ph_1 \dots Ph_n]$ a beszédhangsorozatot, Θ_{Ph_n} az utolsó beszédhanghoz tartozó HMM-et, és $[p_0, p_1, p_2, p_3, p_4]$ az aktuális HMM állapotaihoz tartozó valószínűségi értékeket jelöli. Az egyszerűbb jelölés kedvéért feltettük, hogy a HMM-ek 3 valódi állapottal, és egy kezdő valamint egy végállapottal rendelkeznek.

Két hipotézist szósorozatban ekvivalensnek nevezünk egy időpontban, ha csoportsorozataik és szósorozataik megegyeznek. Két hipotézist egyesíthetőnek nevezünk egy időpontban, ha szósorozatban ekvivalensek, beszédhangsorozataik hasonulástól eltekintve megegyeznek, és az utolsó beszédhangok azonosak (erről a nyelvi modulnak kell tudnia információt adni).

3.2. Hipotézisek kiterjesztései

Hipotéziseket kétféleképpen terjesztünk ki.

1. Hipotézisek akusztikai kiterjesztése. Ekkor a hipotéziseket meghatározó adatok közül csak az időpont, és az aktuális HMM-ek állapotaihoz tartozó valószínűségi értékek változnak.

$$\begin{gathered}\mathrm{H(t, [C_i], [W_i], [Ph_i], \Theta_{Ph_n}, [p_0, p_1, p_2, p_3, p_4])} \\ \Downarrow \\ \mathrm{H'(t+1, [C_i], [W_i], [Ph_i], \Theta_{Ph_n}, [-\infty, p'_1, p'_2, p'_3, p'_4])}\end{gathered}$$

   Itt az új $\mathrm{p'_i}$ értékek a régiekből a szokványos Viterbi algoritmussal [7] számíthatók.
2. Hipotézisek nyelvi kiterjesztése. Ebben az esetben több új hipotézist is kaphatunk, hiszen egy beszédhangsorozat többféleképpen folytatódhat. Az új hipotézisek adatai közül az időpont nem változik, a csoportsorozat, szósorozat, beszédhangsorozat a nyelvi modell által visszaadott módon bővülhet, az aktuális HMM azonosító a nyelvi modell által megadott következő beszédhang szerint változik, és az aktuális HMM állapotaihoz tartozó valószínűségi értékek felveszik kezdeti értéküket.

$$\begin{gathered}\mathrm{H(t, [C_i], [W_i], [Ph_i], \Theta_{Ph_n}, [p_0, p_1, p_2, p_3, p_4])} \\ \Downarrow \\ \left\{ \begin{array}{c} \mathrm{H'(t, [C_i][C'_1], [W_i][W'_1], [Ph_i][Ph'_1], \Theta_{Ph'_1}, [p'_{1_0}, -\infty, -\infty, -\infty, -\infty])} \\ \vdots \\ \mathrm{H'(t, [C_i][C'_N], [W_i][W'_N], [Ph_i][Ph'_N], \Theta_{Ph'_N}, [p'_{N_0}, -\infty, -\infty, -\infty, -\infty])} \end{array} \right\}\end{gathered}$$

Itt $\mathrm{p'}_{k_0} = p_4 + \log(NyelviPont(k))$, és a szósorozat, illetve a csoportsorozat nem feltétlenül bővül. A nem kezdőállapotok valószínűségei felveszik a $-\infty$ kezdeti értékeket (a logaritmikus aritmetika miatt, $log(0)$). Nyelvi kiterjesztés nyilvánvaló feltétele az, hogy $p_4 > -\infty$, tehát az eredeti hipotézis aktuális HMM-jének végállapotához egy nullánál nagyobb valószínűség tartozzon.

3.3. A keresési tér felépítése és vágása

Először megadjuk az általunk használt algoritmus vázát (1. táblázat), majd részletesen kifejtjük az egyes eljárások működését.

Inicializálás(), $t = 0$
H_t=NyelviHipotézisKiterjesztés(KezdőHipotézis)
Ismételd
Jellemzők=BeszédjelJemmezővektorSzámítás()
H_acc=AkusztikaiHipotézisKiterjesztés(H_t, Jellemzők)
H_grm= NyelviHipotézisKiterjesztés(H_acc)
H_{t+1}= HipotézisekEgyesítése(H_acc, H_grm)
$t = t + 1$
Vége

1. táblázat. A hipotézisteret felépítő algoritmus váza

Jellemzővektor-számítás: a beszédfelismerésben szokásos Mel Frequency Cepstral Coefficients (MFCC), energia, Δ, $\Delta\Delta$ jellemzőket tartalmazó 39 dimenziós vektor kiszámítása (alapesetben) 10 ms-onként történik [2].

Akusztikai hipotézis-kiterjesztés: a jellemzővektornak megfelelően, Viterbi kiértékeléssel [7] módosítja a hipotézisek aktuális HMM-jének állapotaihoz tartozó valószínűségi értékeket. A következő algoritmus határozza meg, hogy melyik hipotéziseket terjesztjük ki:

- A hipotézisket rendezzük $max(p_0, \ldots, p_4)$ érték szerint csökkenő sorrendbe.
- Terjesszük ki az első hipotézist, legyen az új hipotézis H1, és az új való-színűségi értékek $p'_0, \ldots, p'_4$. Vegyük az $M = max(p'_0, \ldots, p'_4) - AkusztikaiKüszöb$ értéket, ahol az ***AkusztikaiKüszöb*** egy előre definiált konstans érték.
- Terjesszük ki sorrendben a többi hipotézist is, legfeljebb ***MaxAkusztikai-Kiterjesztés*** számút. Az új hipotézist eldobjuk, ha a hozzá tartozó valószínűségi értékekre: $max(p_0, \ldots, p_4) < M$.

Nyelvi hipotézis-kiterjesztés: lekérdezzük a hipotézisek lehetséges folytatásait, és azokkal új hipotéziseket hozunk létre. Itt a következőképp járunk el:

- A hipotézisket rendezzük a p_4 értékük szerint csökkenő sorrendbe.
- Kiterjesztünk legfeljebb ***MaxNyelviKiterjesztés*** számú hipotézist.
- További feltételként, megállunk a kiterjesztéssel, ha az új hipotézisek száma elérte a ***MaxÚjNyelviHipotézis*** számot.

– Harmadikként, vannak olyan hipotézisek, amelyek teljesen végigmondott szavakhoz tartoznak. Az ilyen hipotézisekből legfeljebb ***MaxSzóvég*** számú hipotézist terjesztünk ki.

A hipotézisek egyesítése: a kétféle kiterjesztéssel kapott hipotézisek között gyakran ekvivalens hipotézispárok jönnek létre. Például, az "a l m" hipotézist kiterjeszthetjük "a l m a"-ra, azonban ez a kiterjesztés már korábban is megtörténhetett, így már létezhet egy "a l m a" hipotézisünk. Az ekvivalens hipotézisek ismétlődését azonban mindenképpen el kell kerülnünk. Mivel a rendszerünkben a HMM-eknél szokásos Viterbi kiértékelést használjuk, ezért az ekvivalens hipotézisek összevonása egzakt módon megtehető a következő, egyszerű módon. Tegyük fel, hogy a nyelvi kiterjesztés után rendelkezünk egy olyan H_n hipotézissel, ami egyesíthető (ld. 3.1. fejezet) egy, az akusztikai kiterjesztés után kapott H_a hipotézissel. Az egyesített hipotézist a következőképpen kapjuk:

$$\begin{gathered}\mathrm{H_n(t,[C_i],[W_i],[Ph_i],}\Theta_{\mathrm{Ph_n}}\mathrm{,[p_0,-\infty,-\infty,-\infty,-\infty])}\\ +\\ \mathrm{H_a(t,[C_i],[W_i],[Ph_i],}\Theta_{\mathrm{Ph_n}}\mathrm{,[-\infty,p_1,p_2,p_3,p_4])}\\ \Downarrow\\ \mathrm{H(t,[C_i],[W_i],[Ph_i],}\Theta_{\mathrm{Ph_n}}\mathrm{,[p_0,p_1,p_2,p_3,p_4])}\end{gathered}$$

Tehát, az egyesített hipotézis – a HMM Viterbi kiértékelésének megfelelően – megegyezik a H_a hipotézissel, de a p_0 értékét a H_n hipotézistől kapja. Az egyesítés során szintén megadunk egy – keresési teret vágó – paramétert, ami az egyesített hipotézisek maximális száma (***HalomMéret***).

Nyelvi hipotézis-kiterjesztés korlátozása: egy további - keresési teret szűkítő - paraméterrel szabályozni tudjuk a nyelvi kiterjesztés gyakoriságát: megadhatjuk, hogy hány iterációnként történjen nyelvi kiterjesztés (***NyelviIterSzám***). Ennek a paraméternek az alapértéke 1, azonban az értékét növelve a be-szédfelismeréshez felhasznált processzoridő nagymértékben csökkenthető (a pontosság némi romlása mellett).

4. Kiértékelés és eredmények

Mielőtt elemeznénk a mérési eredményeket, röviden írunk a rendszerünk implementációjáról algoritmikus szemszögből nézve. A hipotézisek folyamatos kiterjesztése nálunk egy saját fejlesztésű multistack rendszerben valósul meg. Az egyes stack-ek megvalósítása ún. "hybridlist" [8] adatszerkezetre épül, aminek az oka az, hogy olyan változó méretű rendezett adatszerkezetre volt szükségünk, amelynél a keresés, beszúrás és a törlés $O(logN)$ időkomplexitású művelet. Ilyen adatszerkezetek közül - tudomásunk szerint - az említett algoritmus a leggyorsabb. A beszédfelismerő sebessége szempontjából még egy fontos momentum, hogy egy saját memóriakezelőt is fejlesztettünk a minél hatékonyabb memóriahasználat céljából.

4.1. Tanítás és tesztelés

Az akusztikai modul tanításához a Magyar Referencia Beszédadatbázist (MRBA [9]), egy nagyméretű, szegmentált, 332 beszélő által bemondott hanganyagot tartalmazó adatbázis használtuk. A beszédkorpuszban meglévő beszédhangokat 33 csoportra osztottuk fel a tanításhoz. Csoportonként egy-egy három állapotú és állapotonként három Gauss-komponenssel rendelkező balról-jobbra strukturájú HMM-et tanítottunk.

A tesztek során végig szó 3-gram nyelvi modellt használtunk. Ezen modellek létrehozásához egy pajzsmirigy-szcintigráfiás leletekből álló szövegkorpuszt használtunk. Ebből a szövegkorpuszból három, különböző méretű szótárral rendelkező nyelvtant hoztunk létre (500, 1200, 1900 szóalak), mindegyikük részhalmazként tartalmazta a tesztadatbázis bemondásainak szóalakjait (azaz nem volt szótáron kívüli szó teszteléskor).

A tesztelés egy olyan adatbázison történt, amely 100 szcintigráfiás orvosi lelet bemondását tartalmazza (3 nőtől és 2 férfitól), és összesen mintegy 1000 mondatot tesz ki. A tesztek során AMD Athlon 2 GHz processzorral és 2 GB memóriával rendelkező számítógépet használtunk.

A kísérletek során processzoridő-igényt és felismerési pontosságot mértünk. A processzoridő-igényt a táblázatokban másodpercben, valamint Real-Time Factorban (RTF) adjuk meg, amely a felhasznált processzoridő és az összes hanganyag időtartamának (5325 mp.) a hányadosa. A szószintű felismerési pontosságot (accuracy, vagy word recognition rate, WRR) a szokásos eljárással mértük.

4.2. Felismerési pontosság és processzorigény különböző vágásoknál

Elsőként azt vizsgáltuk meg, hogy hogyan függ a felismerési pontosság és a processzorigény a ***MaxÚjNyelviHipotézis*** paraméter értékétől (ami az új hipotézisek maximális számát adja meg nyelvi kiterjesztéskor). A 2. táblázatban foglaltuk össze a tesztek edményeit, melyek során 500 szavas szótárat használtunk 260 küszöbértékű akusztikai vágás mellett. Az eredmények szerint kielégítő eredményt ad (időben és pontosságban is), ha 200-400 új hipotézisben maximáljuk a nyelvi kiterjesztés eredményét. A táblázat alapján – eleinte – ha megduplázzuk a vágási paramétert, akkor hozzávetőlegesen a felére csökken a szófelismerés hibája, azonban később a pontosság nem növelhető tovább. A szótár méretétől függően azt a beállítást célszerű választani, amikor már lényegesen nem változik a hiba, de a rendszer még valós időben működik.

Max. Új Ny. H.	50	100	200	400	600	800	1000
Pontosság	68.8%	84.6%	94.0%	97.0%	97.9%	97.7%	97.7%
Idő (RTF)	0.08	0.14	0.30	0.74	1.20	1.55	1.74

2. táblázat. Az nyelvi kiterjesztés során kapott hipotézisek számára adott korlát hatása a pontosságra és a processzoridő-igényre

A ***NyelviIterSzám*** paramétert – amellyel azt szabályozhatjuk, hogy menynyi iterációnként történjen nyelvi kiterjesztés – a ***MaxÚjNyelviHipotézis*** paraméterrel együtt vizsgáltuk. A 3. táblázat szerint a felhasznált processzoridő folyamatosan csökken, ahogy a nyelvi kiterjesztés (és ezzel együtt a hipotézisek egyesítésének művelete) ritkábban fut le. Azonban az is látszik, hogy a 200/1 -es eset időben jobb, mint a 400/2-es, azaz átlagosan jobban megéri inkább gyakran kevesebb hipotézist kiterjeszteni, mint ritkábban, de többet. Ez a kijelentés viszont csak az öt ember bemondásait tartalmazó tesztadathalmazon számolt átlagos eredményekre igaz. A mi tapasztalataink szerint sokszor megéri a nyelvi kiterjesztést ritkábban elvégezni, például olyan esetekben, amikor a beszélő jól artikuláltan és normál, vagy lassú tempóban beszél (két megfelelő tesztsze-mélyre ld. 4. táblázat). Egy másik eset, amikor jól alkalmazható ez a fajta vágás (ezt most nem vizsgáljuk), amikor a beszédjelből a 10 ms-os alapértéknél sűrűbben nyerünk ki jellemzővektorokat. Ebben az esetben a nyelvi kiterjesztést nem célszerű azonos gyakorisággal végrehajtani.

# Új Ny. H.	200			400			600		
Kit. Iter.	1	2	3	1	2	3	1	2	3
Pontosság	94.0%	88.8%	78.0%	96.4%	93.6%	84.8%	96.3%	94.0%	85.8%
Idő (RTF)	0.30	0.18	0.15	0.64	0.35	0.28	0.75	0.41	0.32

3. táblázat. Eredmények a nyelvi kiterjesztés gyakoriságának és az új hipotézisek maximális számának különböző értékei mellett.

Bemondó	Személy 1.		Személy 2.	
# Max. Új Ny. H.	200		200	
Nyelvi Kit. Iter.	1	2	1	2
Pontosság	98.2%	96.2%	97.4%	97.0%
Idő (RTF)	0.29	0.17	0.30	0.19

4. táblázat. Tesztesetek, amikor a ritkább nyelvi kiterjesztés hatékonynak bizonyult

Egy fontos kérdés lehet az is, hogy mekkorának célszerű választani a ***MaxÚjNyelviHipotézis***, illetve a ***HalomMéret*** paramétereket különböző méretű szótárral rendelkező nyelvtanok esetében. A 5. táblázat tartalmazza az ezzel kapcsolatos méréseinket a (HalomMéret=2· MaxÚjNyelviHipotézis beállítás mellett). Látható, hogy a szótár méretének növelésekor célszerű a halom méretét és a kiterjesztések számát növelni, de a nem feltételez egyenes arányosságot, hiszen az 500-as hipotézisszám korlátozás megfelelő felismerési pontosságot ad az 1900 szavas szótár mellett is. Sajnos az is kitűnik, hogy egy 1000 méretű halommal

és 500 eleműre korlátozott nyelvi kiterjesztéssel már kilépünk a valósidejűség tartományából. Ennek az okáról később részletesebben írunk.

Szótár méret	500		1200		1900	
# Max. Ny. H.	po.	idő	po.	idő	po.	idő
250	95.3%	0.39	90.9%	0.48	85.8%	0.52
500	97.5%	1.13	95.9%	1.26	93.5%	1.33
1000	98.3%	3.62	97.4%	4.22	96.0%	4.49

5. táblázat. A nyelvi kiterjesztést korlátozó paraméter vizsgálata különböző méretű szótárak mellett

A következő vágási paraméter, amelyet vizsgáltunk, az ***AkusztikaiKüszöb***, amely egy alsó korlátot ad meg a hipotézisek pontértékére vonatkozóan. A teszteket 500 szavas szótáron, 250 számúra korlátozott nyelvi kiterjesztés mellett végeztük. A 6. táblázatban közölt eredmények szerint egy 200-250 közötti paraméterérték processzorhasználat és pontosság szemponjából is kielégítő.

4.3. A részműveletek processzoridő-igénye

Ebben az alfejezetben azt hasonlítjuk össze, hogy hogyan változik a 1. tábálázatban megadott alapmetódusok processzoridő-igénye, ha változtatjuk az ***AkusztikaiKüszöb*** (AK), illetve a ***MaxÚjNyelviHipotézis*** (MUH) paramétereket (ld. 7. táblázat). Az itt vizsgált műveletek tehát: jemmezővektor-számítás (JSZ), akusztikai hipotézis kiterjesztése (AHK), nyelvi hipotézis kiterjesztése (NYHK), és a hipotézisek egyesítése (HE).

Akusztikai Küszöb	100	150	200	250	300
Pontosság	7.5%	56.4%	93.7%	95.3%	95.2%
Idő (RTF)	0.03	0.13	0.26	0.38	0.46

6. táblázat. Az akusztikai küszöb hatása

A következőket állapíthatjuk meg a táblázat alapján. A jellemzővektor-számítás időigénye viszonylag stabil, és elhanyagolható mértékű. Az akusztikus küszöb növelésével mind a nyelvi kiterjesztés mind az akusztikai kiterjesztés időigénye megnő (ez érthető, hiszen több hipotézis marad, ami több nyelvi kiterjesztési lehetőséget jelent). A nyelvi kiterjesztéskorláttal egyenes arányban szintén nő mind a nyelvi kiterjesztés mind az akusztikai kiterjesztés időigénye. A hipotézisek egyesítésének időigénye viszont a kiterjesztések időigényénel lényegesen jobban nő. Ennek az a magyarázata, hogy ha mindkét egyesítendő hipotézishalmaz elemszáma megduplázódik, akkor az egyesítéskor egy négyszeres műveletigény fog jelentkezni.

AK	MUH	teljes idő (mp.)					relatív idő			
		Össz.	JSZ	AHK	NYHK	HE	JSZ	AHK	NYHK	HE
200	250	1410	74	408	203	724	5%	29%	15%	51%
250	250	2017	73	526	243	1173	4%	26%	12%	58%
300	250	2433	76	620	266	1471	3%	25%	11%	61%
300	500	6022	78	1021	533	4390	1%	17%	9%	73%
300	1000	19286	88	2100	1176	15922	0%	11%	6%	83%

7. táblázat. A részműveletek időigénye különböző vágási paraméterezéseknél

4.4. A nyelvi modul relatív súlyának és a szótár méretének szerepe

A nyelvi modul (a mi esetünkben szó 3-gram) általában mindenféle szósorozat bemondását megengedi (tehát a szótár bármelyik szava után bármelyik következhet). Azonban azok a szósorozatok, amelyek az adott modell szerint nem lehetségesek (például szó N-gram esetén egy szó N-es egyszer sem fordult elő a tanító szövegkorpuszban) egy megfelelően kicsi nyelvi pontértéket kapnak (jelölje ezt ϵ). Az, hogy ez az érték mennyire kicsi, meghatározza, hogy mekkora lesz a nyelvi modul súlya az akusztikai modulhoz viszonyítva, a nyelvi és akusztikai pontozás kombinálásakor. Egy beszédjel-keret átlagos akusztikai súlya (helyesen felismert bemondásokon mérve) kb. $10^{-20} \dots 10^{-22}$, ehhez viszonyíthatjuk a nyelvi modellünk pontozását. Nyilvánvaló, hogy minél kisebb az ϵ pontérték, annál inkább igazodni fog a felismert szósorozat a nyelvi modellhez (egyszerűen azért, mert amelyik hipotézis nem felel meg a nyelvi modellnek, az egyre inkább lejjebb kerül a rangsorban). A 8. táblázatban foglaltuk össze az eredményeinket. Azt vehetjük észre, hogy minél kisebb az ϵ értéke, annál gyorsabban történik a beszédfelismerés. Ennek az oka az, hogy a nyelvi modell pontozása elnyomja az akusztikai pontozást, és a küszöb szerinti vágás után a hipotézisekből egyre kevesebb marad. Mindemellett a felismerési pontosság egy érték alatt (kb. 1e-40) csökkenni kezd, ami szintén az előbbi magyarázatnak tudható be.

Szótár méret	500		1200		1900	
nyelvi súly	po.	idő	po.	idő	po.	idő
1e-10	87.3%	0.64	74.4%	0.72	65.8%	0.77
1e-20	94.2%	0.53	86.9%	0.62	82.0%	0.68
1e-30	95.5%	0.45	90.7%	0.53	85.7%	0.58
1e-40	95.3%	0.39	90.9%	0.48	85.8%	0.52
1e-50	94.8%	0.36	90.5%	0.42	85.1%	0.47
1e-60	93.9%	0.33	89.1%	0.39	83.6%	0.44

8. táblázat. A nyelvi modul súlyának szerepe különböző méretű szótárak esetén

5. Konklúzió

Ebben a cikkben először leírtuk az általunk fejlesztett folyamatos diktálórendszer leglényegesebb technikai megoldásait. Ezután megvizsgáltuk, hogy a legfon-

tosabbnak vélt, keresési teret vágó paramétereknek mi a szerepük, és hogyan befolyásolják a processzoridő-igényt, valamint a felismerési pontosságot. Arra a következtetésre jutottunk, hogy a nyelvi modell súlyának, illetve az akusztikai küszöb értékének egy elég stabil optimális értéke van, függetlenül a szótár méretétől. Ezzel szemben nagyobb szótár használatakor a nyelvi kiterjesztést korlátozó, illetve a halomméretet megadó paraméter értékének a növelésével tudunk csak megfelelő felismerési pontosságot elérni. Szerencsére az ezzel kapcso-latos kisérletek azt mutatták, hogy nem szükséges ezek lineáris növelése. Az eredmények arra is rávilágítottak, hogy ha nagyszótáras rendszer felé akarunk később továbblépni, akkor a jelenlegi, hipotézisek egyesítését végző algoritmusunkat le kell cserélnünk egy hatékonyabbra, mert időigény szempontjából ez a kritikus része a diktálórendszerünknek.

Hivatkozások

1. Chou, W., Juang, B.H., eds.: Pattern Recognition in Speech and Language Processing. CRC Press, Inc., Boca Raton, FL, USA (2002)
2. Rabiner, L., Juang, B.H.: Fundamentals of speech recognition. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1993)
3. Ravishankar, M.: Efficient algorithms for speech recognition (1996)
4. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing. Prentice Hall (2001)
5. Bánhalmi, A., Kocsor, A., Paczolay, D.: Magyar nyelvű diktáló rendszer támogatása újszerű nyelvi modellek segítségével. In: MSZNY. (2006) 337–347
6. Forney, G.D.: The viterbi algorithm. Proceedings of The IEEE **61** (1973) 268–278
7. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. (1990) 267–296
8. Kozub, D.: http://www.dankozub.com/cpp/hlist.htm (1999)
9. Vicsi, K., Kocsor, A., Teleki, C., Tóth, L.: Beszédadatbázis irodai számítógép-felhasználói környezetben. In: MSZNY. (2004) 315–318