Kulcsszókeresési kísérletek hangzó híranyagokon beszédhang alapú felismerési technikákkal

Gosztolya Gábor[1], Tóth László[1]

[1]MTA-SZTE Mesterséges Intelligencia Tanszéki Kutatócsoport,
Szeged
{ggabor, tothl}@inf.u-szeged.hu

Kivonat: A beszédadatbázisok kereshetővé tételéhez szöveges címkékkel kell azokat ellátni. A kézenfekvő megoldás szószintű átirat készíttetése lenne nagyszótáras beszédfelismerővel. A felismerők azonban zárt szótárral dolgoznak, így előfordulhat, hogy számunkra fontos keresési kifejezéseket (tulajdonneveket, névelemeket) esélyünk sem lesz megtalálni, pusztán mert azok nem szerepelnek a felismerő szótárában. Jelen cikkben olyan megoldásokat hasonlítunk össze, amelyek csupán beszédhang szinten végzik el az előzetes indexálást, így tetszőleges keresési kifejezésre (hangsorozatra) képesek rákeresni. A vizsgált módszerek találati pontossága gyakorlati szempontból is használhatónak ígérkezik, köszönhetően az eleve magas beszédhang-felismerési pontosságnak. A futási időt tekintve azonban még a leggyorsabb módszer is sokkal lassabbnak bizonyul, mint ami egy ilyen alkalmazástól elvárt lenne. Ezért a későbbiekben kifinomult indexálási technikák bevetésére lesz szükség.

1 Bevezetés

A beszédadatbázisok kereshetővé tételének legkézenfekvőbbnek tűnő módja egy beszédfelismerő lefuttatása a hanganyagon: ez elvileg szöveges átiratot készít a felvételekről, amelyeken ezután már a hagyományos szöveges keresési és indexálási módszereket alkalmazhatjuk. A gyakorlatban azonban az általános célú, nagyszótáras felismerők még elég nagy hibaaránnyal dolgoznak (magyar nyelvre 80% körüli szópontosság a legjobb ismert eredmény [13]). Nyilvánvaló módon a hibásan felismert szavakat a szöveges keresés során biztosan elveszítjük. Ezen a problémán lehet valamelyest segíteni oly módon, hogy nemcsak a legvalószínűbb átiratot generáltatjuk le a felismerővel, hanem ún. „N-best” kimenetet készítünk, amelyben a bizonytalan pontokon több lehetséges illeszkedő szó is szerepel (de ezzel a keresési időt és a „vakriasztás” esélyét is növeljük). A hibásan felismert szavak mellett van azonban egy másik, kevésbé nyilvánvaló probléma is a fent leírt technológiával: az, hogy a beszédfelismerők mindig egy zárt szótárral dolgoznak, így a szótárukban nem szereplő szavakat soha nem fogják megtalálni. A zárt szótár problematikája legfőképpen a főnevek, azon belül is a tulajdonnevek, illetve tágabban véve a névelemek kategóriáját érinti: ezek azok a szófajok, amelyeken belül folyamatosan keletkeznek új szavak,

A kutatást részben az NKTH TÁMOP-4.2.2/08/1/2008-0008 programja támogatta.

vagy legalábbis előtérbe kerülnek olyan szavak, amelyek a szótár összeállításakor nem forogtak közszájon, s így a szótárba sem kerültek be. Viszonylag újonnan keletkezett köznévre lehet példa a „teljesítményvolumen-korlátozás", névelemre pedig egy újonnan bejegyzett cégnév, pl. „Sokasara Kft.". A háttérből előbukkanó, majd újra eltűnő tulajdonnév esetét példázza Biszku Béla neve, amely egy hétig naponta szerepelt a híradókban, előtte viszont évekig nem, és azóta ismét nem. A beszédfelismerők szótárát, illetve nyelvi modelljét statisztikai úton, nagyméretű szöveges korpuszok alapján automatikusan szokás összeállítani. Amennyiben tehát egy szó vagy névelem nem fordult elő a tanítókorpuszban – akár mert még nem létezett, akár mert „lappangott" –, akkor az a szó nem kerül be a felismerő szótárába, és így felismerni sem fogja tudni azt. Az ilyen szavakat OOV – „out of vocabulary" – névvel illeti a szakirodalom. Egy alaposan összeállított nyelvi modell mellett ezek az OOV szavak viszonylag ritkák, így például egy diktálási feladatnál csak kevés hibát okoznak. Teljesen más azonban a helyzet, ha nem diktálásról, hanem hanganyagokban való keresésről van szó. Kereséskor ugyanis jóval gyakrabban írunk be főneveket, illetve névelemeket, mint amilyen azoknak a természetes szövegekben való előfordulási gyakorisága. A Yahoo vizsgálatai szerint a webes keresőjükbe beírt keresési kifejezések 70%-a főnév, amelynek több mint fele (40%) tulajdonnév [2]. A Microsoft hasonló elemzése szerint a keresések 71%-a tartalmaz névelemet [4], egy harmadik tanulmány szerint a webes keresések 11-17 százaléka irányul személynévre [1]. Mindez azt mutatja, hogy kereséskor pont azok a szavak fontosak, amelyeknél a legnagyobb a kockázata annak, hogy a beszédfelismerő nem ismeri őket. Természetesen elvileg megoldható a felismerő nyelvi modelljének folyamatos bővítése, ekkor azonban a teljes felismerést is újra és újra le kell futtatni, ami nehézkes és nagyon időigényes.

Az OOV szavak elkerülésére a felismerő előzetes lefuttatásával szemben elvileg lehetséges az a megoldás is, hogy a felismerőt csak a keresési kulcsszó megadása után futtatjuk le, természetesen csak az adott szóra. Nagyon nagy adatbázis esetén azonban ez nem járható út, mert még az egyetlen szóval történő teljes felismertetés is túl sokáig tart. Olyan megoldást kell tehát találnunk, amely bizonyos szintig elvégzi a felismerést, de a szószintű valószínűségek kiszámításánál hamarabb megáll. Ennek egyik módja lehet, ha nem szavakkal indexáltatjuk a beszédkorpuszt, hanem annál kisebb egységekkel, például beszédhangokkal. A felhasználó által beadott keresési kifejezést tehát a felismerő beszédhang szintű kimenetében fogjuk keresni. Sajnos ennek a módszernek is megvan a maga hátránya: a beszédhang-felismerési pontosságok a szószintű pontosságnál jóval rosszabbak, általában 50-70% közé esnek. Emiatt tehát egy hibákkal erősebben terhelt kimenetben kell keresnünk, és a környezet (szószint) sem segít, emiatt magas lesz például a vakriasztások száma rövid szavak esetén. A keresés maga is bonyolultabb, és emiatt lassabb lesz, mint szószintű címkézés esetén. Az irodalomban emiatt a szóalapú és a beszédhang alapú technikák kombinált használatát tartják a legjobbnak [6].

Jelen cikkben többféle beszédhang alapú keresési technikát hasonlítunk össze híradófelvételekben való keresési feladaton. A híradós felvételeknek csak a hírközlő által bemondott blokkjaiban keresünk, azaz alapvetően jó hangminőségű és szépen artikulált beszédről van szó, aminek köszönhetően viszonylag magas, 80% fölötti beszédhang-felismerési pontosságot sikerült elérnünk. A cikkben bemutatjuk magát a beszédkorpuszt, valamint a felismerésben használt neuronhálós technológiát. Ezután

kiindulási alapként két olyan kulcsszókereső módszert is kipróbálunk, amelyek a neuronháló adatkeret szintű kimenetén dolgoznak, tehát a lokális valószínűségek letárolásától eltekintve gyakorlatilag a teljes felismerést lefuttatják az adott kulcsszóval. Mint említettük, ez a megoldás viszonylag lassú, ezért kipróbálunk egy olyan algoritmust, amely a felismerő által kiadott N-best beszédhanghálóban dinamikus programozással, a tévesztési mátrixot figyelembe vevő szerkesztési távolság alapján keresi meg a kulcsszó feltételezett előfordulásait. A negyedik algoritmus pedig csak a felismerő által kiadott legvalószínűbb fonémasorozatot dolgozza fel, így várhatóan pontatlanabb, de gyorsabb, mint a teljes hálón dolgozó megoldás.

2 A felhasznált beszédadatbázis és feldolgozása

A kulcsszódetektálási kísérletekhez 70 híradót rögzítettünk nyolc tévécsatornáról (ATV, Hálózat TV, Hír TV, M1, M2, RTL, Tv2). A felvételeket néhány mondatos blokkokra vágtuk és három kategóriába soroltuk minőség szerint: a „tiszta" kategóriába kerültek azok a felvételek, amelyeken szépen artikulált beszéd hallható, és a háttérzaj minimális. Tipikusan ide tartoznak a stúdióban, a műsorvezetőktől elhangzó részletek. „Zajos" besorolást kaptak a tervezett beszédet tartalmazó, de magasabb zajszintű felvételek – jellemzően a külső helyszínen tartózkodó riporterek bejelentkezése. Végezetül, a „spontán" címkét kapták a spontán beszédet tartalmazó blokkok – ezek tipikusan a riportalanyok szájából elhangzó mondatok. Jelen cikkben csak a tiszta besorolású felvételeket használtuk fel, abból kiindulva, hogy többnyire minden hír előtt elhangzik egy stúdiós felvezető, így ezek kereshetővé tételével a teljes hírblokkot is meg lehet találni. A 70 híradót 44-9-17 arányban osztottuk fel betanítási (train), fejlesztési (development) és tesztelő (test) blokkokra, ügyelve arra, hogy a tévécsatornák mindegyikéből kerüljön mindegyik részhalmazba. Időtartamot tekintve kb. 5 és fél óra - 1 óra - 2 óra arányban oszlanak el a felvételek a blokkok között. A felvételek mindegyikét legépeltük, az ortografikus átiratot utólagosan is ellenőriztük. A leiratok fonetikus átiratát egy egyszerű fonetikus átíróval készítettük el, amely csak egyetlen átiratot rendel minden szóhoz, és csak egyszerű hasonulási szabályokat használ.

3 Nagy pontosságú beszédhang-felismerés neuronhálókkal

Kísérleteinkben a beszédfelismerőt szószintű nyelvi modell nélkül fogjuk futtatni, azaz pusztán a fonetikai szintű kimenete alapján szeretnénk a keresendő kifejezések előfordulásait megtalálni. Ezért érthető módon rendkívül sok múlik azon, hogy a fonetikai kimenet mennyire pontos. Angol nyelvre a TIMIT beszédadatbázison végezték a legtöbb beszédhang szintű felismerési kísérletet, és az eredmények azt mutatják, hogy neuronhálós technikákkal jobb eredményeket lehet elérni ezen a téren, mint a hagyományos rejtett Markov-modelles (HMM) megoldásokkal [11, 7]. Ezért kísérleteinkben mi is egy neuronhálót használtunk a beszédjel adatkereteinek fonetikai címke-valószínűségekkel való ellátására. A későbbiekben bemutatandó

kulcsszókereső algoritmusok egy része közvetlenül ezeket a keretszintű valószínűségi értékeket használja fel. Más részük viszont beszédhang szintű felismerési kimenetet, azaz beszédhang-sztringet vagy hálót igényel bemenetként. A felismerés lefuttatásához a neuronháló kimenete integrálható a hagyományos rejtett Markov-modellbe, így kapjuk az úgynevezett hibrid HMM/ANN rendszereket [3]. Mi a közismert HTK rendszert [18] módosítottuk úgy, hogy képes legyen a neuronháló által nyújtott lokális valószínűségekből felismerést végezni. Ily módon a hagyományos (Gauss-görbékkel dolgozó) és a hibrid modell közvetlen összehasonlítására is lehetőségünk nyílt.

A beszédjelek fonetikai címkézésére, illetve a keresendő kulcsszavak fonetikai átírására 52 címkét használtunk, ezek lényegében megfelelnek a magyar nyelv hangkészletének. A beszédtechnológiában az akusztikus modellezésben azonban igen elterjedt megoldás az úgynevezett környezetfüggő vagy trifón modellek használata. Ennek lényege, hogy a címkézést tovább finomítjuk oly módon, hogy az egyes hangok különböző hangkörnyezetben előforduló változatai különböző címkéket kapnak. Ezzel megkönnyítjük az algoritmusok számára a címkék elkülönítését, így a felismerési pontosság javul. A módszer hátránya, hogy a modellek száma megnő, így a tanítás és a felismerés is lassabb lesz, és a trifón címkék kezelése speciális problémákat is okoz. Szerencsére a HTK csomag fel van készítve a trifón modellek készítésére, így alapmodellként egy monofón és egy trifón HMM modellt is tanítottunk a korábban ismertetett adatbázison. Akusztikus jellemzőként a szokványos 13 mel-kepsztrális (MFCC) együtthatót használtuk, azok első és második deriváltjaival. Az egyes beszédhangokhoz háromállapotú modelleket rendeltünk, a HTK trifón-készítő algoritmusa ezeket 1073 környezetfüggő állapotra („szenonra”) képezte le. Nyelvi modellként csak egy szimpla beszédhangbigramot alkalmaztunk. A monofón, illetve trifón modellekkel kapott beszédhang-felismerési pontosságokat az 1. táblázatban láthatjuk.

A neuronháló betanításához adatkeret szintű fonetikai címkézésre van szükség, ezt az előzőekben betanított HMM-ek segítségével, kényszerített illesztéssel állítottuk elő. Háromféle címkézéssel is kísérleteztünk: az egyik esetben a monofón címkét rendeltük minden kerethez, azaz az 52 címke egyikét. A második esetben a monofón modell állapotának azonosítójára tanítottunk, ez esetben 3*52=153 elemből állt a címkekészlet. Végezetül, a harmadik kísérletben a trifón modell állapotazonosítóit taníttattuk a hálóval, ez esetben az osztályok száma 1073 volt. A rejtett neuronok száma minden esetben 5000-re volt állítva, és egyszerű packpropagation tanítást végeztünk. Inputként a hálózat 9 egymás melletti MFCC-vektort használt.

Viszonylag új felfedezés, hogy a neuronháló pontossága tovább javítható, ha a kimeneteire egy újabb hálót tanítunk [10, 5]. Ez a második háló a kontextus segítségével képes az első háló hibáin javítani, és részben a hangkapcsolatok modellezését is átveszi a bigram nyelvi modelltől. Ezt a technikát „kétfázisú” megoldásnak fogjuk nevezni a továbbiakban. Az 1. táblázatban soroltuk fel a különböző neuronháló-címkézési és -tanítási stratégiákkal kapott modellek beszédhang-felismerési pontosságát. Az eredmények megerősítik azt a korábbi megfigyelésünket [16], hogy a monofón tanítású hibrid körülbelül olyan teljesítményre képes, mint a hagyományos trifón HMM. Ha pedig a neuronhálót a trifón címkékhez igazítjuk, akkor további jelentős pontosságnövekedést tudunk elérni. A legjobb rendszer 83%-os pontossága alapján bíztunk abban, hogy a kulcsszavak megtalálása is lehetséges lesz pusztán a

fonetikai kimenet alapján. A következő fejezetben az e célból bevetett algoritmusokat ismertetjük.

1. táblázat: Beszédhang-felismerési pontosság a különböző akusztikus modellekkel.

Akusztikus modell	Pontosság
HMM, monofón	67,18%
HMM, trifón	75,38%
Hibrid, monofón, 1 állapot	75,56%
Hibrid, monofón, 3 állapot	76,93%
Kétfázisú hibrid, monofón, 1 állapot	77,46%
Kétfázisú hibrid, monofón, 3 állapot	79,18%
Kétfázisú hibrid, trifón	83,33%

4 Kulcsszókeresési megközelítések

A kulcsszókeresési probléma egy információ-visszakeresési (*information retrieval*, IR) feladat: adott hanganyagban keressük egy adott kulcsszóhalmaz előfordulásait. Egy kulcsszókeresési eljárás tehát találatok egy listáját adja vissza, melyek jellemzően rendelkeznek valószínűséggel is, mely alapján rangsorolhatók. Ezt a rangsort azonban, néhány más információ-visszakeresési problémával ellentétben, nem a felhasználónak adott felsorolás sorba rendezésére használjuk, hanem a találatok további szűrésére (melyet pl. a FOM kiértékelési metrika is kihasznál, l. 5.1 alfejezet).

Érdemes megjegyeznünk, hogy az angol nyelvű szakirodalomban a probléma megnevezésére két kifejezés is elterjedt: *keyword spotting*, illetve *spoken term detection* (STD). A legtöbb szerző egyetért abban, hogy vannak különbségek a két terület között, de abban már nem, hogy pontosan mik is ezek: a legkorábbi különbségtétel szerint a keyword spotting magában a hanganyagban keres, míg az STD valamilyen köztes reprezentációban. Más források szerint a fő eltérés a kulcsszavak szótárának zártságában (keyword spotting) vagy nyíltságában (STD) van; végül különbséget szokás tenni a pontosság mérésére szolgáló metrikák használata alapján is (keyword spotting: FOM, spoken term detection: ATWV; részletesebben l. 5.1 alfejezet) [17]. Az angol terminológia bizonytalansága miatt mi egységesen a kulcsszókeresés kifejezést használjuk.

A kulcsszókeresési problémában jelenleg több megközelítésnek is van létjogosultsága. Mivel a keresett kulcsszóval ellentétben a felvételek, amelyekben keresünk, előre ismertek, azok feldolgozását bizonyos mértékig előre elvégezhetjük. Az egyes megközelítések közötti alapvető különbség az, hogy a teljes feldolgozás mekkora része történik ebben az előkészítő szakaszban. A feldolgozás bizonyos lépései a keresett kulcsszó ismerete nélkül csak közelítőleg végezhetőek el; amennyiben ezeket is az előkészítő szakaszhoz soroljuk, azzal a keresési rész időigényét csökkentjük, azonban egyúttal információt is veszítünk, mely könnyen vezethet a pontosság kisebb-nagyobb mértékű csökkenéséhez. A következőkben áttekintjük a legelterjedtebb megközelítéseket.

4.1 Viterbi-keresés

A jelenlegi beszédfelismerési technikákhoz legközelebb álló megközelítésben a keresett kifejezést a beszédfelismerésben megszokott módon, a keretszintű valószínűségeket aggregálva próbáljuk ráilleszteni a bemondásokra. Találatot akkor jelzünk, ha az illeszkedés valószínűsége egy bizonyos küszöb fölé esik. Ekkor tehát az előkészítő részbe soroljuk a bemondásokon végzett szokásos műveleteket egészen a jellemzőkinyerési és fonémavalószínűség-számítási fázisokig; ezek után a keretszintű fonémavalószínűségeket tároljuk el.

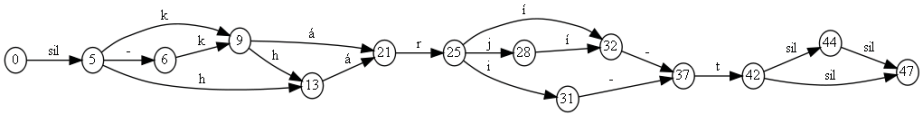
A megközelítés előnye, hogy szinte kizárólag a beszédfelismerésben szokásos technikákat alkalmazza, melyek azon a területen már hatékonynak bizonyultak. Ugyanígy, a későbbiekben a beszédfelismerés területén bevezetett pontosságnövelési technikák is könnyen átvehetők. Hátránya viszont egyrészt a bemondásokhoz eltárolandó adatmennyiség nagysága, másrészt a keresés nagy műveletigénye.

Ebben a cikkben ezt a megközelítést két konkrét implementációval valósítottuk meg; az első egy dinamikus táblatöltéses eljárás, mely az egyes keretszintű fonémavalószínűségeket kombinálja össze. A keresés végeredménye azon nem átfedő hipotézisek listája, melyek (fonémaszámmal normált) valószínűsége egy bizonyos küszöb fölé esik. Algoritmusunk futási ideje két okból is lényegesen magasabb az optimálisnál: egyrészt implementációja Matlabban történt, másrészt a konkrét megvalósításban minden kulcsszóra teljesen külön keresést végzünk azok párhuzamosítása helyett.

Alternatív megoldásként a HTK beszédfelismerő rendszerrel [18] is készítettünk egy kulcsszókereső modellt. Mivel egy beszédfelismerőnek minden jelszakaszhoz kell outputot adnia, ezért kulcsszókeresésre úgy használható fel, ha a szótárába a kulcsszavak mellé ún. „filler” vagy „garbage” elemeket veszünk fel; mi a legegyszerűbb megoldásként a beszédhangmodelleket használtuk ilyen célra. Ilyenkor a rendszer beszúrási büntetésének állításával hangolhatjuk be a találatok és a vakriasztások arányát.

4.2 Hálóalapú keresés

Az előző megközelítés két jelentős hátrányán (nagy eltárolt adatmennyiség, időigényes keresés) javíthatunk, ha további lépéseket sorolunk át az előkészítő szakaszba. Erre a legelterjedtebb megoldás a hálóalapú keresés: ennek során a keretszintű valószínűségeken fonéma N-gram nyelvtant használva hagyományos beszédfelismerést végzünk, majd a talált legjobb hipotézisek gráfját tároljuk el, a keresendő kifejezéseket pedig erre a hálóra (*lattice*) illesztjük (l. 1. ábra). E megoldás kétségtelen előnye, hogy a bemondásonként eltárolandó adatmennyiség lényegesen kevesebb, valamint – a háló méretétől függően – a keresés is gyorsabb lehet. Hátránya viszont, hogy nagymértékben támaszkodik a fonémafelismerés pontosságára: az ekkor elkövetett hibákat a későbbi lépésekben már nehezen vagy egyáltalán nem lehet korrigálni. Mivel az, hogy a keresett kifejezés összes fonémáját hibátlanul azonosítsuk, igen valószínűtlen, a keresés során bizonyos büntetésekkel beszúrást, törlést és csere műveletet is meg szokás engedni (bár ez lassítja a keresést). Emellett, a fonémafelismerés hibáit ellensúlyozandó, annak tévesztési mátrixa alapján fonémánként eltérő büntetősúly rendelhető a beszúrás és törlés műveletekhez, fonémapáronként eltérő pedig a cseréhez.



1. ábra. Példa fonémahálóra; a csúcsok időpontoknak felelnek meg, az élek címkéi az adott szakaszhoz rendelt fonémák. „sil" a csendet, „-" a zárhangok zárfázisát jelöli.

Cikkünkben ezt a megközelítést egy külső rendszer használatával képviseltettük: a Brnói Műszaki Egyetem LatticeSTD rendszerét használtuk [12]. Az előfeldolgozást és az előkészítő szakaszt a HTK [18] rendszerrel végeztük, a hálót fonéma 2-gram nyelvtant használva generáltattuk le, a háló méretét – a HTK rendszer megfelelő programjának, a HVite-nek N-best paraméterét használva – 3-ra állítottuk.

4.3 Legvalószínűbb fonémasorozaton alapuló keresés

A legvalószínűbb találatok hálóban tárolása még mindig elég bonyolult adatreprezentációt igényel, melyben a keresés is időigényes. A bemondás eltárolt modelljének további egyszerűsítésével mindkét hátulütőn javíthatunk. Kézenfekvő egyszerűsítés, ha csak a beszédfelismerési keresés során legvalószínűbbnek talált fonémasorozatot tároljuk el (természetesen a fonémák közti határok helyével és az egyes fonémaelőfordulások valószínűségével együtt) a legjobb hipotéziseket leíró háló helyett. Ennek egyértelmű előnye a reprezentáció egyszerűsége, amely a keresési algoritmus időigényét is csökkenti. Az egyszerű adatformátum lehetővé teheti olyan reprezentáció használatát is, mellyel a keresés nagymértékben felgyorsítható (indexálás). További előny, hogy az eltárolt adatok mennyisége a hálóban tároltnál egy nagyságrenddel kisebb. A megközelítés hátránya viszont, hogy a beszédfelismerési keresés során szuboptimálisnak bizonyult utak elhagyásával információt veszítünk, és még a hálóalapú keresésnél is nagyobb mértékben hagyatkozunk a fonémaosztályozóra.

Ezt a megközelítést is egy saját implementációjú kereső módszer (egy dinamikus táblatöltési eljárás) képviseli. A hálóban kereséshez hasonlóan itt is megengedünk beszúrást, cserét és törlést is, melyeket szintén a fonémafelismerés tévesztési mátrixából számolunk. A keresés végeredménye azon nem átfedő hipotézisek listája, melyek (fonémaszámmal normált) valószínűsége egy bizonyos küszöb fölé esik. Ennek a módszernek az implementálása is Matlabban történt, így futási ideje mindenképpen magasabb, mint egy gépközeli (C++, Java) megvalósításé lenne, emellett itt is az egyes kulcsszavak keresése a többitől függetlenül történik.

5 Tesztelés és eredmények

A kulcsszókeresési probléma és az alkalmazott algoritmusok leírása után most rátérünk a tesztkörnyezet ismertetésére: bevezetjük a pontosság- és sebességmérésre

használt metrikákat, vázoljuk a tesztelés menetét, végül prezentáljuk és elemezzük az elért eredményeket.

5.1 Kiértékelési metodikák

A kulcsszókeresési probléma egy információ-visszakeresési probléma, emiatt hagyományos IR-metrikákkal: pontossággal (*precision*) és fedéssel (*recall*) is mérhető egy adott algoritmus-konfiguráció teljesítménye [15, 17]. A legtöbb információ-visszakeresési területen a két metrikát azok (parametrikus) harmonikus közepével, az F-mértékkel (*F-measure*) szokás egyetlen értékké aggregálni, azonban a kulcsszókeresés területén más metrikák terjedtek el. Leggyakrabban a Figure-of-Merit (FOM) mérőszámot használják, mely az óránként és kulcsszavanként 1, 2, … 10 hibás találat megengedése esetén elért fedési értékek számtani közepe. A másik elterjedt mérőszámot az amerikai National Institute of Standards and Technology (NIST) vezette be 2006-os kulcsszókeresési versenyén: ez az aktuális kulcsszósúlyozott érték (*Actual Term-Weighted Value*, ATWV), mely a következőképpen definiált:

$$\mathrm{ATWV} = 1 - \frac{1}{T}\sum_{i=1}^{T}\left(P_{Miss}(t) + \beta P_{FA}(t)\right), \tag{1}$$

ahol $P_{Miss}(t)$ az adott kulcsszó eltévesztésének, $P_{FA}(t)$ pedig hibás találatának valószínűsége; azaz

$$P_{Miss}(t) = 1 - \frac{N_{corr}(t)}{N_{true}(t)} \quad \text{és} \quad P_{FA}(t) = 1 - \frac{N_{FA}(t)}{T_{speech} - N_{true}(t)}, \tag{2}$$

ahol $N_{corr}(t)$ az adott kulcsszó helyes találatainak, $N_{true}(t)$ a valós előfordulásainak, $N_{FA}(t)$ a hamis találatainak száma, T_{speech} pedig az átfésülendő felvételek összhossza másodpercben [8, 9]. β értéke általában 1000. Egy tökéletesen működő rendszer ATWV pontszáma 1,0, egy olyané, amely egyáltalán nem ad vissza találatokat, 0,0. Feltételezve, hogy T_{speech} lényegesen nagyobb, mint $N_{true}(t)$, egy olyan rendszer, amely az összes előfordulást megtalálja, de minden kifejezésre óránként 3,6 hamis találatot produkál, szintén 0,0 értéket fog kapni, így ez a metrika jóval szigorúbb, mint a FOM. További különbség, hogy az ATWV az összes visszaadott találatot figyelembe veszi, míg FOM esetén csak a valószínűbbeket (melyek megtartásával a hamis találatok száma még óránként és kulcsszavanként 1, 2, …, 10 alatt marad). A tesztek során elsősorban az ATWV metrikát alkalmaztuk, de az adott konfigurációhoz tartozó FOM pontszámot is feltüntettük.

Az egyes módszerek teljesítménye mellett fontos tényező azok futási ideje is. Ezt általában egyórányi hanganyagra és egy kulcsszóra vetített, másodpercben mért időigényben szokás megadni [9, 17], így mi is ezt az utat követtük.

5.2 A tesztelés menete

A tesztelést 25 darab, 2-6 szótagos kulcsszóval végeztük, melyek kellő számban fordultak elő az adatbázisban, és amelyeket valószínű keresési kifejezéseknek ítéltünk. Figyelembe véve a magyar nyelv agglutináló voltát, azt is helyes találatnak értékeltük, amennyiben a szövegben előforduló szó teljes egészében tartalmazza a keresett kulcsszót, vagy annak olyan alakját, melyben a szóvégi magánhangzó meghosszabbodott (pl. az „Amerika" kulcsszó esetén az „Amerikában" is helyes találat). A híradós felvételekből körülbelül egyórányi anyagot (a fejlesztési részt) a rendszerek fejlesztése, paramétereik finomhangolása során vettünk igénybe; az így optimalizált keresőeljárások teljesítményét pedig a mintegy kétórányi tesztfelvételhalmazon mértük le.

A módszerek hatékonyságának mérésére az ATWV metrikát alkalmaztuk. Mivel a két saját módszer esetében a találati listára akkor veszünk fel egy lehetséges találatot, ha valószínűsége egy bizonyos határ fölött van, ennek a küszöbnek a kiválasztása sem triviális; ezt úgy tettük meg, hogy minden lehetséges küszöbértékre kapott listára kiszámoltuk az ATWV metrikát a felvételek fejlesztési részén, és azt a küszöböt választottuk, amely a maximális értékhez vezetett. Ezek után a végső tesztfelvételhalmazon ezt a küszöböt alkalmazva számítottuk ki az ATWV értéket. Az érdekesség kedvéért feltüntettük az MTWV metrikát is: ezt úgy kapjuk, hogy az összes lehetséges küszöbérték használatával megszűrt találatlistára kiszámítjuk az ATWV-t, majd vesszük ezek maximumát. Mivel ezt a tesztfelvételekre tettük meg, ez lényegében azt mutatja meg, hogy optimálisra választott küszöb esetén mekkora ATWV értéket érhetnénk el.

Harmadikként kiszámítottuk a FOM százalékot is. Megjegyzendő, hogy a HTK és az általunk tesztelt, hálóban kereső módszer (LatticeSTD) elég szűk találati listát ad vissza: a hamis riasztások száma gyakran az óránként és kulcsszavanként 2-t sem éri el, míg a valós FOM érték meghatározásához szükséges az összes találat megadása felvételóránként és kulcsszavanként 10 hamis riasztásig; ebből következően ennél a két rendszernél a feltüntetett FOM százalékok csak tájékoztató jellegűek, a valós pontszám ezeknél feltehetően valamivel (2-3 százalékponttal) magasabb.

Az eljárások futási idejét az adatbázis tesztelésre fenntartott részén mértük (egy 3,0 GHz-es Intel Core2 Duo számítógépen 2GB RAM-mal), és egyórányi felvételre és egy kulcsszóra igénybe vett másodpercben fejeztük ki. Azon rendszerek esetében, melyek futási ideje függ a paraméterbeállítástól is (HTK, LatticeSTD), csak olyan paramétereket használtunk, melyekkel a lefutást még „kivárhatónak" ítéltük.

5.3 Eredmények

A 2. táblázatban láthatók az egyes módszerek által elért eredmények. Az előkészítő fázis során kipróbáltuk mind az egyszerűbb (egyállapotú) monofón, mint a bonyolultabb, de pontosabb trifón modellt; az utóbbi modelleket a Viterbi-keresés általunk tesztelt implementációja már nem tudta kezelni, így ezt az eljárást csak monofón modellel próbáltuk ki. (A fennmaradó módszerek közül a HTK képes trifónokkal is dolgozni, a hálóalapú és a legjobb fonémasorozatban kereső módszerek esetén pedig

ez a kérdés csak az előkészítő szakaszt érinti, maguk a keresőeljárások már csak az 52 önálló fonémát tartalmazó hálót, illetve sorozatot kapják meg.)

2. táblázat: Az egyes keresési megközelítések teszteredményei monofón és trifón fonémamodellt alkalmazva.

	Monofón fonémamodell			Trifón fonémamodell		
Keresési módszer	ATWV	MTWV	FOM	ATWV	MTWV	FOM
Viterbi	0,54	0,60	89,98%	–	–	–
HTK	0,52	0,58	90,42%	0,62	0,63	88,93%
Hálóalapú (LatticeSTD)	0,48	0,48	73,24%	0,65	0,65	82,64%
Legjobb fon.sorozatban	0,46	0,52	85,03%	0,43	0,58	88,37%

Az eredmények alapján a módszerek már a gyakorlatban is használható pontosságot adnak. Az is látható, hogy trifón modellt használva lényegesen javulnak az egyes módszerek teljesítményei: a növekedés jóval nagyobb, mint amit a fonémaszintű pontosság 77,46%-ról 83,33%-ra emelkedésétől várnánk, ami valószínűleg annak tudható be, hogy a tárgyalt kulcsszókeresési módszerek alapvetően a fonémaosztályozóra hagyatkoznak. Az egyes megközelítések teljesítményét összevetve jól látható, hogy ahogy egyre több részfeladatot helyezünk át az előkészítő szakaszba, és ennélfogva egyre kevesebb információ alapján végezzük a keresést, úgy csökken a keresés hatékonysága. A 3. táblázatból (mely az egyes megközelítések időigényét tartalmazza másodpercben, egy kulcsszóra és egy órányi hanganyagra vetítve) azonban az is kiviláglik, hogy mindez együtt jár a keresési idő jelentős csökkenésével is. Megjegyzendő, hogy a HTK-val és a hálóalapú keresőrendszerrel ellentétben (melyek C++-ban íródtak) a két saját eljárás Matlabban íródott, így nem igazán futási időre optimalizált. További hátrányuk, hogy a kulcsszavak keresése egymástól függetlenül történik, míg például a HTK rendszer a 25 kulcsszót párhuzamosan illesztette a bemondásokra. Valószínűleg ez a magyarázata a HTK kiugró gyorsaságának monofón fonémamodell esetén, azonban a gyakorlatban ritka az a szituáció, mikor egynél több kifejezést keresnénk párhuzamosan.

3. táblázat: Az egyes keresési megközelítések (előkészítő szakaszt nem tartalmazó) keresési időigénye másodpercben, egy kulcsszóra és egyórányi hanganyagra.

Keresési módszer	Monofón modell	Trifón modell
Viterbi	122,29	–
HTK	2,25	21,30
Hálóalapú (LatticeSTD)	34,91	34,87
Legjobb fonémasorozatban	10,75	10,75

Látványos az ATWV és MTWV értékek szignifikáns különbsége a Viterbi és a legjobb fonémasorozatban keresés módszerek esetén, míg ez a differencia a HTK rendszernél (trifón esetben legalábbis) igen kicsi, a LatticeSTD esetében pedig nulla. Ebből arra következtethetünk, hogy a két saját módszernél az egyik adatbázison (a

fejlesztési részen) megállapított küszöb nem stabil, más felvételhalmazon (esetünkben a tesztelési részen) jóval az optimális alatt teljesít, ami felveti valamilyen más küszöbszámítási módszer (pl. a keresett kulcsszó fonémaszáma helyett a találat időtartama alapján normalizálás) szükségességét.

Összességében azt mondhatjuk, hogy míg az egyes rendszerek pontossága már eléri a gyakorlati felhasználás szintjét, időigényük még meghaladja a tolerálható mértéket. Például egyhónapnyi híradófelvételben egyetlen kulcsszó megtalálása a leggyorsabb eljárásnak is majdnem három percébe kerülne, ami egy átlagos felhasználó szempontjából egyértelműen túl sok. Emiatt további programozási és indexálási trükköket szoktak bevetni, amelyek további részeredményeket leszámolva növelik ugyan a tárigényt, de gyorsítják a visszakeresést. Például a szerkesztési távolság számítása (a cserélési, törlési és beszúrási lehetőségek végigvizsgálata) gyorsítható oly módon, hogy az összes lehetséges (max k. hosszú) részstring távolságát előre kiszámítjuk és eltároljuk [14].

6 Konklúzió

Cikkünkben egy nagypontosságú fonémafelismerőre alapozva különféle kulcsszókeresési algoritmusokat hasonlítottunk össze. Várakozásainknak megfelelően azt találtuk, hogy ha a számítások egyre nagyobb részét toljuk át az előkészítő fázisba, annál gyorsabb lesz ugyan a keresés, de a pontosság is egyre romlik. Mindezzel együtt is úgy véljük, hogy az elért pontosságértékek gyakorlatilag is használhatóak lehetnek – a futási időkön viszont feltétlenül csökkenteni kell. Ezért további fejlesztésként különféle kifinomult indexálási technikák bevetése lenne a legfontosabb. Szerencsére jelenleg ez nagyon aktív kutatási terület, és az irodalom számos megoldást kínál erre a problémára. További érdekes kutatási lehetőség lenne a nagyobb egységekkel (pl. szótagok) történő indexálással való kiegészítés, valamint – mint bevezetőnkben említettük – a szószintű rendszerekkel való kombinálás, hiszen az általunk javasolt módszerek főleg az OOV szavak esetén ígérnek jelentős javulást.

Bibliográfia

1. Artiles, J., Gonzalo, J., Sekine, S.: WePS 2 Evaluation Campaign: Overviews of the Web People Search Clustering Task. In: Proc. WWW 2009 (2009)
2. Barr, C., Jones, R., Regelson, M.: The Linguistic Structure of English Web-Search Queries. In: Proc. EMNLP (2008)
3. Bourlard, H., Morgan, N.: Connectionist Speech Recognition – A Hybrid Approach. Kluwer (1994)
4. Guo, J., Xu, G., Cheng, X., Li, H.: Named Entity Recognition in Query. In: Proc. SIGIR (2009)
5. Ketabdar, H., Bourlard, H.: Enhanced phone posteriors for improving speech recognition systems. IEEE Trans. ASLP, Vol. 18, No. 6 (2010) 1094–1106
6. Mamou, J., Mass, Y., Ramabhadran, B., Sznajder, B.: Combination of multiple speech transcription methods for vocabulary independent search. In: Proc. SIGIR (2008)

7. Mohamed, A.-R., Dahl, G., Hinton, G.: Deep Belief Networks for phone recognition. In: Proc. NIPS 22 workshop on deep learning for speech recognition (2009)
8. NIST: The Spoken Term Detection (STD) Evaluation Plan. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA. http://www.nist.org/speech/tests/std (2006)
9. Pinto, J., Szöke, I., Prasanna, S.R.M., Hermansky, H.: Fast Approximate Spoken Term Detection from Sequence of Phonemes. In: Proc. SIGIR (2008) 28–33
10. Pinto, J. et al.: Analysis of MLP based hierarchical phoneme posterior probability estimator. IEEE Trans. ASLP, megjelenés alatt (2010)
11. Siniscalschi, S.M., Schwarz, P., Lee, C.-H.: High-accuracy phone recognition by combining high performance lattice generation and knowledge-based rescoring. In: Proc. ICASSP (2007) 869–872
12. Szöke, I., Schwarz, P., Matejka, P., Karafiát, M.: Comparison of Keyword Spotting Approaches for Informal Continuous Speech. In: Proc. Interspeech (2005)
13. Tarján, B., Mihajlik, P., Tüske, Z.: Nagyszótáras híranyagok felismerési pontosságának növelése morfémaalapú, folyamatos beszédfelismerővel. In: MSZNY (2009) 185–194
14. Thambiratnam, K., Sridharan, S.: Rapid Yet Accurate Speech Indexing Using Dynamic Match Lattice Spotting. IEEE Trans. ASLP, Vol. 15, No. 1 (2007) 346–357
15. Tikk, D. (szerk.): Szövegbányászat. Typotex, Budapest (2007)
16. Tóth. L., Tarján, B., Sárosi, G., Mihajlik, P.: Speech Recognition Experiments with Audiobooks. Acta Cybernetica, megjelenés alatt.
17. Wang, D.: Out-of-Vocabulary Spoken Term Detection. PhD thesis, University of Edinburgh (2010)
18. Young, S.J. et al: The HMM Toolkit (HTK) (software and manual). http://htk.eng.cam.ac.uk/ (1995)