

**Szegedi Tudományegyetem
Mesterséges Intelligencia Kutatócsoport**

**Osztályozók kombinációs rendszerei
és alkalmazásuk
a természetes nyelvi technológiában**

PhD-értekezés tézisei

Felföldi László

Témavezető:

**Prof. Dr. Csirik János
Dr. Kocsor András**

**Szeged
2008**

Bevezető

A mintaillesztő rendszerek tervezésének célja olyan osztályozó készítése, amely a rendszer viselkedését hűen modellezi[8; 20; 30]. Az osztályozó egy minta-tér bármely eleméhez hozzárendel egy osztálycímktét a lehetséges címkék véges halmazából. Egy osztályozó konstruálása (tanítása) során ismert példákból általánosítja az optimális hozzárendelést, aminek hatékonysága így erősen függ a tanító mintatér és az adott példák karakterisztikájának előzetes ismeretétől. Végtelen mennyiségű tanító adat felhasználásával minden konzisztens tanulóalgoritmus a Bayes-féle optimális osztályozást állítaná elő. A gyakorlatban azonban mindig csak véges és zajos mintahalmaz áll rendelkezésünkre, így a különböző eljárások szükségszerűen különböző osztályozáshoz vezetnek. Ezért érdemes egy adott probléma megoldása érdekében több különböző algoritmust is betanítani, ekkor nagyobb eséllyel garantálhatjuk, hogy jó modellt, illetve paraméterbeállítást kapunk. Ezek közül választhatjuk valamilyen szempont szerint a legjobbat, azonban egy ilyen választás csak a legritkább esetben optimális, hiszen értékes információt veszíthetünk a többi osztályozó kimenetének elhanyagolásával. Ez az információvesztés elkerülhető, ha az összes rendelkezésünkre álló osztályozó algoritmus kimenetét figyelembe vesszük a kombinációs módszerek segítségével.

A szakirodalomban több kombinációs technika vizsgálata megtalálható[7; 20; 21]. Az egyszerű statikus stratégiák, mint a "Prod", "Sum", "Min", "Max" és "Borda Count" szabály, melyeket gyakran alkalmaznak összetett osztályozó rendszerekben, egyszerűségüknek köszönhetően igen nagy népszerűsége tettek szert, azonban nem lehet őket konkrét alkalmazások speciális igényeihez igazítani. Másrészt, az Adaboost algoritmus és változatai hatékonyan képesek javítani az eredeti osztályozó teljesítményét, különösen az ún. "gyenge" osztályozók esetén, viszont az eljárás több száz vagy akár több ezer iterációs lépést is igényelhet. Gyakorlati alkalmazások esetében legtöbbször nem áll rendelkezésünkre kellő mennyiségű erőforrás ilyen hatalmas mennyiségű osztályozás kiértékeléséhez, amely ezen módszerek használhatóságát erősen korlátozza. Ebben a disszertációban egy hatékony kombinációs eljárást mutatunk be, amely jó teljesítményt biztosít efféle alkalmazások során is.

Az automatikus beszéd felismerő rendszerek és alkalmazások általában valamilyen gépi tanulási eljárást alkalmaznak - mint például a mesterséges ideghálózatok (ANN) vagy a Gauss keverék modellek (GMM) - a beszédjel szegmenseinek vagy kereteinek címkézésére. A fonéma-klasszifikáció minőségének javítása így hatással van a teljes rendszer működésére is, ezért érdemes megvizsgálnunk, hogy a különböző kombinációs stratégiák milyen hatással lehetnek az alkalmazás felismerési teljesítményére.

A kombinációs technikák egy másik biztató alkalmazási területe a természetes nyelvi feldolgozás. A szófaji egyértelműsítés (POS tagging) és a főnévi csoportok azonosítása (NP parsing) esetében az egyszerű kombinációs stratégiák, mint például a többségi szavazás (Majority Voting) vagy a szorzás szabály (Prod Rule) hatékonyan bizonyultak a rendszer teljesítményének növelésére. A fejlettebb adaptív technikák azonban eredeti formájukban nem használhatóak környezetfüggő alkalmazásokban. Ebben a disszertációban megvizsgáljuk az adaptív stratégiák olyan általánosítását, amely képes a természetes nyelvi feldolgozás speciális igényeinek kezelésére.

Többsztályozós rendszerek

Amennyiben több osztályozó áll rendelkezésünkre, a legegyszerűbb módszer a rendszer teljesítményének növelésére, hogy kiválasztjuk azt az osztályozót, amely egy adott adatbázison valamilyen szempontból

a legjobban viselkedik. Az osztályozás során csak ennek az osztályozónak a kimenetét számítjuk ki, és csak ezt az információt vesszük figyelembe a végső döntés meghozatalakor. Ez a módszer felfogható egy „korai” kombinációs technikának, és a gyakorlati alkalmazása igen széleskörű.

Azonban egy ilyen választás a legritkább esetben optimális, hiszen értékes információt veszíthetünk a többi osztályozó kimenetének elhanyagolásával. Ez az információvesztés elkerülhető, ha az összes rendelkezésünkre álló osztályozó algoritmus kimenetét figyelembe vesszük a kombinációs módszerek segítségével.

Az osztályozók információ-típusai

Az osztályozó-algoritmusok kombinációjának célja, hogy egy adott mintához egy osztálycímét rendeljen, felhasználva az egyes osztályozók által szolgáltatott kimenetek együttesét. Definíciójuk szerint minden osztályozó módszer alkalmas arra, hogy az általa legjobbnak vélt osztálycímét megadja, azonban számos gépi tanulási algoritmus képes ennél több információt is nyújtani, amelyeket a kombinációs módszerek felhasználhatnak. A kombiációs stratégiák a következő típusú osztályozókat különböztetik meg az általuk szolgáltatott információ alapján:

- **Absztrakt információ:** Az osztályozó csak a mintához rendelt osztálycímét szolgáltatja. A kombinációs módszerek azon csoportjához, amelyeknek elég ez a típusú információ, tartoznak a szavazáson alapuló technikák, mint például a Bagging és a Boosting.
- **Sorrendi információ:** A legjobb osztály címkéje helyett az osztályozó az osztálycímek valószínűségük szerint rendezett listáját tudja szolgáltatni. Ennek az információt feldolgozó kombinációs technikának egyik képviselője a „Borda Count” kombinációs módszer.
- **Mérték vagy bizonyossági információ:** A legáltalánosabb esetben az osztályozó minden osztály a posteriori valószínűségét képes szolgáltatni. A kombinációs technikák ezen valószínűségeket összesítik valamilyen módon, és a legnagyobb kombinált a posteriori valószínűségű osztályt választják eredményül. Ilyen kombinációs módszerekre példa a „Prod”, a „Sum” és a „Max” szabályok által generált kombinációk.

Statikus Kombinációs Sémák

A következőkben feltesszük, hogy az osztályozó algoritmusok bizonyossági információt képesek szolgáltatni. Tekintsünk egy mintaillesztési problémát, ahol az \mathbf{x} mintát kell hozzárendelnünk a minták $(\omega_1, \dots, \omega_M)$ halmazához. Tegyük fel, hogy R osztályozó áll rendelkezésünkre, ezek mindegyike használhat külön jellemzővektort a minta leírására. Jelölje $\mathbf{x}^{(i)}$ az \mathbf{x} minta \mathcal{C}_i tanuló által használt reprezentációját. A jellemzőtérben minden ω_k osztály modellezhető egy $p(\mathbf{x}^{(i)}|\omega_k)$ valószínűségi eloszlásfüggvénnyel és $P(\omega_k)$ a priori valószínűséggel. A Bayes-elmélet alapján az adott $\mathbf{x}^{(i)}$ $i \in \{1, \dots, R\}$ jellemzők esetén az \mathbf{x} mintát a legnagyobb a posteriori valószínűségű ω_j osztályhoz kell rendelnünk:

$$f(\mathbf{x}) = \omega_j, \quad j = \underset{k}{\operatorname{argmax}} P(\omega_k|\mathbf{x}_1, \dots, \mathbf{x}_R).$$

A következőkben felsorolunk néhány, a gyakorlatban használt kombinációs módszert:

- „Product” szabály:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k P(\omega_k) \prod_i p(\mathbf{x}^{(i)} | \omega_k).$$

- „Sum” szabály:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k \left[(1 + R)P(\omega_k) + \sum_i p(\omega_k | \mathbf{x}^{(i)}) \right].$$

- „Max” szabály:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k \left[(1 + R)P(\omega_k) + R \max_i p(\omega_k | \mathbf{x}^{(i)}) \right].$$

- „Min” szabály:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k \left[P^{1-R}(\omega_k) + R \max_i p(\omega_k | \mathbf{x}^{(i)}) \right].$$

- „Median” szabály:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k \operatorname{med}_i p(\omega_k | \mathbf{x}^{(i)}).$$

- „Majority Voting” szabály:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k \sum_i \Delta_{ki},$$

ahol Δ_{ki} a $P(\omega_k | \mathbf{x}^{(i)})$ valószínűség bináris alakja:

$$\Delta_{ki} = \begin{cases} 1 & \text{ha } P(\omega_k | \mathbf{x}^{(i)}) = \max_j P(\omega_j | \mathbf{x}^{(i)}) \\ 0 & \text{egyébként,} \end{cases}$$

- „Borda count”:

$$f(\mathbf{x}) = \omega_j, \quad j = \operatorname{argmax}_k \sum_i \rho_{ki},$$

ahol ρ_{ki} a sorrendhez tartozó mérőszám:

$$\rho_{ki} = \frac{1}{C} \sum_{j: P(\omega_j | \mathbf{x}^{(i)}) \leq P(\omega_k | \mathbf{x}^{(i)})} 1.$$

Additív kombinációs modellek

Habár a statikus kombinációs technikákat egyszerűségük miatt igen gyakran alkalmazzák többosztályozós rendszerekben, nem lehet őket konkrét alkalmazások speciális igényeihez igazítani. Ezért a fejlettebb adaptív módszerek - mint például az additív kombinációs sémák - kerültek a kutatások középpontjába. Ezek a kombinációs technikák az osztályozók eredményeinek súlyozott összege alapján választják ki a kombináció eredményét. Feltéve, hogy az osztályozók bizonyossági információt szolgáltatnak, a kombináció leírható a következő képlettel:

$$\hat{f}_i(x) = \sum_{j=1}^N w_j f_i^j(x).$$

A legjobb kombináció eléréséhez a kombináció paramétereit úgy kell beállítanunk, hogy az eljárás optimálisan viselkedjen egy kiválasztott adatbázison. Az általunk vizsgált lineáris kombinációk alakja igen egyszerű, a tanítás során csak a súlyok helyes értékeit kell meghatároznunk. A következőkben megadunk néhány, a gyakorlatban használt módszert:

1. *Egyszerű átlagolás.* A legegyszerűbb esetben minden együtthatóhoz ugyanazt az értéket rendeljük:

$$w_j = \frac{1}{N}.$$

2. *Súlyozott átlagolás.* A tapasztalatok szerint az egyszerű átlagolásnál jobb eredmény érhető el, ha az együtthatók a megfelelő osztályozó hibarányával fordítottan arányosak:

$$w_j = \frac{1}{E_j},$$

ahol E_j a C_j osztályozó hibaránya, vagyis a tanító adathalmazon helyesen címkézett és az összes minta számának hányadosa.

3. *Hierarchikus módszerek.* A w_j súlyok kiszámításához a számított és a várt a posteriori valószínűségek közötti, valamilyen távolságfüggvénnyel meghatározott eltérést kell minimalizálnunk:

$$\min_w \sum_{x \in \mathcal{X}} \sum_{i=1}^l \mathcal{L}(\hat{f}_i(x), p_i(x)),$$

ahol $\mathcal{L}(\hat{f}_i(x), p_i(x))$ a költség (loss) függvény. Mivel a gépi tanulási algoritmusok felfoghatóak olyan optimalizáló eljárásoknak, amelyek egy adott loss függvény várható értékének egy adott adatbázison való minimumát keresik, a kombináció optimális súlyait egy megfelelő gépi tanulási módszer használatával határozhatjuk meg.

Bagging

A Bagging algoritmus a tanító adatbázis elemeiből generált ún. „bootstrap” adatbázisokon tanított osztályozók szavazatai alapján választja ki a bemenethez tartozó osztálycímkét. A bootstrap adatbázis elemeit a tanító adatbázis példái közül, egyenletes eloszlás szerint véletlenszerűen választjuk ki, egy

elemet akár többször is. A Bagging futtatása során T számú B_1, B_2, \dots, B_T bootstrap adatbázist generál, és a tanulóalgoritmus segítségével minden B_i adatbázison egy C_i osztályozót készít. A kombinált osztályozó C^* a C_1, C_2, \dots, C_T osztályozóból épül fel, kimenete az az osztálycímke, amelyre a legtöbb osztályozó szavaz (majority voting).

Algorithm 1 Bagging algoritmus

Require: Tanító adatbázis S , Osztályozó algoritmus \mathcal{I}

Ensure: Kombinált osztályozó C^*

for $i = 1 \dots T$ **do**

S' = bootstrap minták az S adatbázisból

$C_i = \mathcal{I}(S')$, az osztályozó tanítása

end for

$$C^*(\mathbf{x}) = \operatorname{argmax}_j \sum_{i: C_i(\mathbf{x}) = \omega_j} 1$$

A tanító adatbázis egy adott elemét $1/m$ valószínűséggel választhatjuk ki m elem közül, így annak a valószínűsége, hogy az elem legalább egy példányban bekerüljön a generált bootstrap adatbázisba, $1 - (1 - 1/m)^m$. Nagy m esetén ez a szám körülbelül $1 - 1/e = 63,2\%$. Ez az oka annak, hogy nem stabil osztályozó algoritmus (ANN, döntési fák) esetén az így tanított osztályozók különbözőek lesznek, és ezek kombinálásával a rendszer hatékonysága növelhető. Azonban a Bagging algoritmus leronthatja a stabil osztályozó-algoritmusok teljesítményét, mivel az egyes osztályozók tanításhoz kevesebb független adat áll rendelkezésre.

Boosting

A Boosting alapötlete, hogy egyszerű osztályozók iteratív kombinációjával építsünk fel egy hatékony osztályozót. Az algoritmus elméleti háttere a PAC (Probably Approximately Correct) tanulás[29] elméletéből eredeztethető. A PAC elmélet formalizmusával megmutatható, hány példa szükséges az algoritmus tanításhoz, hogy az adott valószínűséggel helyes választ szolgáltatson. Vailant megmutatta[29], hogy egyszerű osztályozók kombinációjával, amelyek alig jobbak, mint a véletlen választás, tetszőleges pontosságú rendszer építhető fel.

A legnépszerűbb Boosting módszer az Adaboost (Adaptive Boosting) algoritmus[12]. Az Adaboost adaptív abban az értelemben, hogy az új osztályozó hipotézist az előzőleg választott hipotézisek hibái alapján konstruáljuk meg. A Bagging-gel ellentétben, ez a módszer alkalmas arra, hogy az osztályozó algoritmust a nehéz példák helyes kezelésére kényszerítse. Az adaptív működés a tanítási példák súlyeloszlásának bevezetésével és kezelésével biztosítható. A nagy súllyal rendelkező példák nagyobb hatással kell legyenek az osztályozó hipotézis kiválasztására, mint a kisebb súlyú példák. Minden iterációs lépésben a súlyokat úgy módosítjuk, hogy a rosszul viselkedő példák nagyobb súlyt kapjanak.

Az \mathcal{L} loss függvény, amelyet a súlyok frissítéséhez használunk, tetszőlegesen választható. Az eredeti Adaboost algoritmus az exponenciális költség- (loss) függvényt alkalmazza, azonban más függvényekkel is felépíthető hatékony Boosting módszer, mint például a Logitboost [13] és az Arcing [5].

Algorithm 2 Boosting algoritmus

Require: Tanító adatbázis $Z_N = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$, iterációk száma T

Ensure: Kombinált osztályozó C^*

$$D_n^{(1)} = 1/N \text{ minden } n = 1, \dots, N.$$

for $t = 1 \dots T$ **do**

S' = bootstrap adatbázis Z_N -ből, D_n^t eloszlással.

$$C_t = \mathcal{I}(S')$$

$$\epsilon_t = \frac{1}{n} \sum_{n=1}^N \mathcal{L}_{1/0}(y_n, C(\mathbf{x}_n))$$

ha $\epsilon_t > 1/2$ **akkor** kilépés

Optimális w_t együttható választása

$$D_n^{(t+1)} = \frac{D_n^{(t)} \mathcal{L}(y_n, C(\mathbf{x}_n))}{N_t},$$

ahol N_t normalizációs együttható, hogy $\sum_{n=1}^N D_n^{(t+1)} = 1$ teljesüljön.

end for

$$C^*(\mathbf{x}) = \text{sign}(\sum_{t=1}^T w_t C_t(\mathbf{x}))$$

Analitikus Hierarchikus Folyamat

A lineáris kombinációs technikák, különösen a Boosting algoritmus különböző változatai rövid idő alatt igen nagy népszerűsége tettek szert a gépi tanulási alkalmazásokban, mivel számottevően képesek javítani az osztályozás hatékonyságát. Az Adaboost algoritmus osztályozók egy sorozatát állítja elő oly módon, hogy ugyanazt a tanulóalgoritmust az adatbázis különböző, speciálisan mintavételezett változatain tanítja be. Ezzel a módszerrel – különösen az ún. "gyenge" osztályozók esetében – drasztikus javulást lehet elérni az osztályozó egyszeri tanításához képest, azonban az eljárás több száz vagy akár több ezer iterációs lépést is igényelhet. Gyakorlati alkalmazások esetében legtöbbször nem áll rendelkezésünkre kellő mennyiségű erőforrás ilyen hatalmas mennyiségű osztályozási példa kiértékeléséhez, amely a módszerek használhatóságát erősen korlátozza.

A disszertációban bemutatunk egy hatékony kombinációs megoldást, amely ezen alkalmazások esetében is használható. A módszer az analitikus hierarchikus folyamaton (Analytic Hierarchy Process, AHP), a többtényezős döntések kedvelt algoritmusán alapul, amely a páronkénti összehasonlításal megadott kritériumokhoz tartozó lehető legjobb döntés meghatározására alkalmas.

Fontosságának kiszámításához a kritériumokat páronként hasonlítjuk össze, és egy A összehasonlítási mátrixot építünk. Az A mátrix a_{ij} eleme tartalmazza az i -edik kritérium j -edikhez viszonyított relatív fontosságát. Ebből következik, hogy az a_{ji} elem az a_{ij} elem reciproka. Legyen az y alternatíva fontossága v , amit az alkalmazott kritériumok lineáris kombinációjával fejezünk ki:

$$v(y) = \sum_{j=1}^n w_j v(y_j),$$

ahol w_j az y alternatíva y_j kritériumhoz tartozó fontossági értéke. Az összehasonlítási mátrixok felhasználásával az AHP kiszámítja a hierarchia legfelsőbb kritériumaitól az alternatívákig az összes csomópont fontossági értékét, és a legnagyobb értékhez tartozó alternatívát választja végső döntésként.

Definiáljuk a súlyarányok W mátrixát:

$$w_{ij} = \frac{w_i}{w_j}.$$

Egyszerűen ellenőrizhető, hogy a w vektor a W mátrix sajátvektora, és a hozzá tartozó sajátérték n :

$$\begin{aligned}(Ww)_i &= \sum_{j=1}^n W_{ij}w_j = \\ &= \sum_{i=1}^n \frac{w_i}{w_j}w_j = \sum_{j=1}^n w_i = \\ &= nw_i.\end{aligned}$$

Az AHP célja a w súlyvektor kiszámítása az A páronkénti összehasonlítási mátrixokból, feltéve, hogy az A mátrix elemei a súlyarányok mért vagy becsült értékei:

$$a_{ij} \approx \frac{w_i}{w_j},$$

ahol a fontossági értékek az ismeretlen w vektor elemei.

Saaty ajánlásai szerint feltesszük, hogy $a_{ij} > 0$, valamint

$$a_{ij} = \frac{1}{a_{ji}}.$$

A mátrixelmélet szerint ilyen mátrix elemeinek kis mértékű perturbációja csak kis mértékben módosítja a mátrix sajátértékeit. Így feltehetjük, hogy az n -hez legközelebb eső sajátértékhez tartozó sajátvektor a súlyok jó közelítésének felel meg.

A lineáris kombinációs stratégiákban a kombinált a posteriori valószínűségek az osztályozók kimeneteinek súlyozott összege, vagyis:

$$f_i(x) = \sum_{j=1}^N w_j f_i^j(x).$$

A lineáris kombináció és a többtényezős döntések lineáris alakja közötti hasonlóságot kihasználva megkonstruálhatunk egy újszerű kombinációs módszert, amely az osztályozók teljesítményének páronkénti összehasonlítása esetén az analitikus hierarchikus eljárás segítségével a kombinációban szereplő súlyokat meghatározza.

A tesztek során összehasonlítottuk a különböző algoritmusokkal felépített lineáris kombinációs módszerek hatékonyságát. Megvizsgáltunk két alpmódszert, az SA és WA (egyszerű és súlyozott átlagolás) algoritmusokat, és 4 AHP kombinációs stratégiát, amelyek az összehasonlítási mátrix felépítéséhez használt adatbázisok méretében különböztek egymástól. Az eredmények szerint mindegyik kombinációs módszer javította az osztályozó általánosítási képességét, és majdnem minden esetben az ajánlott AHP alapú stratégia jobb eredményt ért el, mint az alpmódszerek.

Az analitikus hierarchikus folyamatra épülő kombinációs módszer ötletét és az elért eredményeket [10]-ben publikáltuk.

Fonéma-osztályozás

Az automatikus beszéd felismerés az alakfelismerés speciális problémája, amely az emberi beszédérzékelés és beszédértés utánzására törekszik. A beszédjel akkora változatosságot képes mutatni, hogy a beszédtechnológia kutatások a tudás-bázisú beszéd felismerő rendszerek építése helyett inkább a statisztikai

	Set1	Set2	Set3	Set4	Set5
<i>SA</i>	8.70	8.34	7.88	7.26	7.78
<i>WA</i>	7.56	7.64	7.64	7.06	7.68
<i>AHP₁</i>	6.84	7.26	7.04	6.76	7.48
<i>AHP₂</i>	6.74	6.90	6.98	6.82	7.56
<i>AHP₃</i>	6.67	6.94	6.96	6.80	7.58
<i>AHP₄</i>	6.78	7.00	6.88	6.82	7.54

1. táblázat. A kombinációs módszerek osztályozási hibái [%] az UCI Letter adatbázison. (Az osztályozó hibája kombináció nélkül 13,78%)

	Prod	Sum	Max	Min	Borda	Voting
g1set1	12.00%	11.64%	12.59%	12.77%	12.77%	13.23%
g1set2	11.76%	11.41%	12.06%	13.06%	12.06%	11.47%
g1set3	11.70%	11.35%	13.18%	12.29%	11.64%	10.87%
g1set4	12.77%	12.41%	14.24%	12.35%	12.59%	11.41%
g1set5	10.87%	10.70%	11.88%	11.17%	11.41%	11.17%
g2set1	8.63%	8.45%	9.22%	9.10%	8.87%	9.16%
g2set2	8.75%	8.57%	9.87%	9.16%	9.10%	9.04%
g2set3	7.03%	7.09%	8.04%	7.33%	7.80%	7.15%
g2set4	8.98%	7.98%	9.87%	9.34%	8.69%	7.74%
g2set5	8.51%	8.22%	8.98%	7.98%	8.81%	8.51%
g3set1	5.14%	5.14%	6.32%	5.61%	5.61%	5.56%
g3set2	5.02%	5.50%	5.73%	4.85%	5.08%	5.08%
g3set3	5.14%	4.91%	5.26%	5.20%	5.08%	4.91%
g3set4	4.67%	4.61%	5.02%	4.79%	5.02%	4.91%
g3set5	5.02%	4.96%	5.08%	5.14%	5.14%	5.14%

2. táblázat. Osztályozási hibák az ANN, SVM, és kNN osztályozók hibrid kombinációja esetében

mintaillesztésen alapuló technikákat részesítik előnyben. Ebben az esetben azonban rengeteg megoldandó problémával szembesülünk, amelyek közül talán a legfontosabb a felismerési egység kérdése. A statisztikai feldolgozás alapja, hogy osztályok egy véges halmaza esetén nagy számosságú tanító példát felhasználva meghatározzuk az ezekhez tartozó valószínűségi eloszlást. A felismerés során az ismeretlen beszédjel szakaszait kellene ezekhez az osztályokhoz rendelniük valamilyen hasonlósági mérték alapján. Mivel a lehetséges mondatok vagy akár a lehetséges szavak száma is potenciálisan végtelen, valamilyen kisebb, véges számosságú egységet kell választanunk az általános beszéd felismerési feladatokban. A leggyakrabban alkalmazott ilyen egység a fonéma, így a következőkben a fonémák osztályozását fogjuk megvizsgálni.

A kísérleteket a kutatócsoport által kifejlesztett szegmens alapú beszéd felismerő rendszer, az Oasis Speech Lab segítségével végeztük. Az első lépésben a megelőző tesztek során hatékonynak bizonyuló osztályozó algoritmusok (SVN, ANN, kNN) kombinációs eredményeit hasonlítottuk össze. A 2. táblázat tanulsága alapján nincsen egyértelműen legjobb kombinációs szabály, amely ezen osztályozók felhasználásával minden adatbázison optimális lenne. A Sum szabályt használó algoritmusok érték el a legjobb eredményt, de a módszerek hatékonysága között csak igen kicsi a különbség.

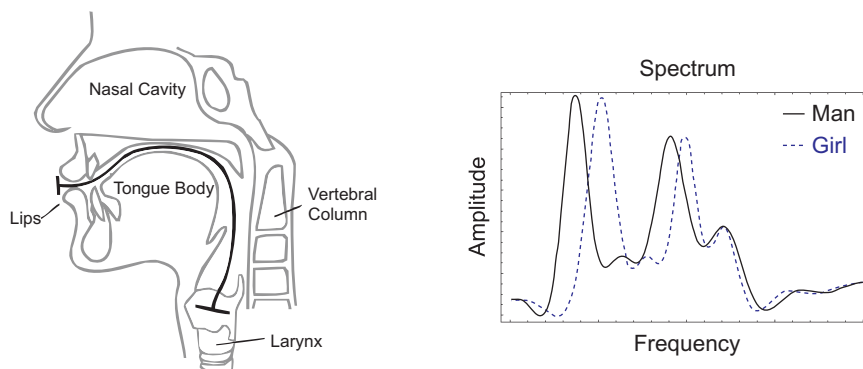
A következő lépésekben a Bagging és Boosting módszer hatékonyságát hasonlítottuk össze. A Bagging algoritmus növelni tudta a felismerés hatékonyságát, eredményei hasonlóak az egyszerű kombinációval elért eredményekhez, azonban ezt jóval több osztályozó felhasználásával érte el. Mivel a Boosting módszer a Bagging továbbfejlesztett alakjának is felfogható, használatával jobb eredményekre számítottunk, azonban ezen az adatbázison a két algoritmus ugyanazt az eredményt érte el. Ennek oka, hogy a felhasznált osztályozó algoritmus „túl erős”, a tanító adatbázison az osztályozási hiba nagyon kicsi. Így az algoritmus következő iterációiban az osztályozónak már gyakorlatilag csak a zajos elemekre kell fókuszálnia, amelyek helyes osztályozása igen nehéz, és a osztályozási hiba könnyen meghaladhatja az 50%-os határt.

A fonémafelismerés kombinációs módszereinek vizsgálatát [11] -ban publikáltuk.

Artikulációs csatorna normalizáció

A beszéd felismerő rendszerek hatékonysága nagyban függ az alkalmazott fonémafelismerő modul minőségétől. Beszédterápiai rendszerek, mint például a „Beszédmester” fonológiai tudatosság-fejlesztő programcsomag esetében ez még inkább megnyilvánul. Mivel a rendszernek megbízhatóan kell működnie mind a gyerekek, mind a különböző korú felnőttek esetében, a fonémafelismerőt alkalmazni kell mindkét nemű és gyakorlatilag minden életkorú beszélő kezelésére. A feladat speciális abból a szempontból is, hogy képes kell legyen izolált fonémák felismerésére, így nem hagyatkozhatunk nyelvi szabályok segítségére. Következésképpen nagy teher nehezedik az akusztikus felismerőre, és bármely módszer, ami a fonémafelismerést javítja, nagy hatással bír a teljes rendszer minőségére.

A lehetséges technikák egyike az artikulációs csatorna hossza szerinti normalizáció (VTLN), amely igen eredményesnek bizonyult olyan rendszerekben, ahol a beszélők nem és életkor szerinti eloszlása igen változatos. Az off-line VTLN algoritmusok [6; 9; 31] alkalmazásával olyan felismerő építhető, amely hatékonyan kezeli mind a férfiak, a nők és a gyerekek által kiejtett hangokat.



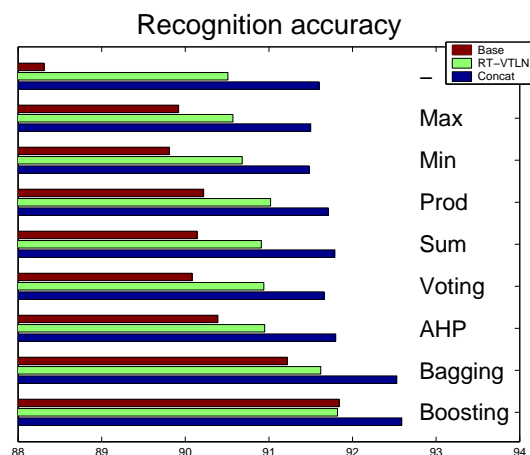
1. ábra. Az artikulációs csatorna, és hatása a szöveg formánszerkezetére. A jobb oldali grafikon ábrázolja egy magánhangzó spektrumát férfi (folytonos vonal) illetve női (szaggatott vonal) beszélő esetén.

Claes mérései szerint[6] az artikulációs csatorna hossza férfiak esetében 17 cm, a nők és gyerekek esetében 15 illetve 14 cm. A formánsfrekvenciák pozíciói az artikulációs csatorna hosszával fordítottan arányosak, ez okozza a tapasztalt eltérést a formánsok szerkezetében.

A normalizáció során a beszédhangok frekvenciáit módosítjuk valamely transzformációs függvény segítségével. Amennyiben az artikulációs csatornát egy L hosszúságú csőként modellezzük, akkor a transzformációs függvény lineáris. A valóságban azonban az artikulációs csatorna jóval komplexebb alakú, amit bonyolultabb transzformációs függvényekkel próbál közelíteni a szakirodalom[9; 31].

Adott transzformációs függvény esetén a normalizáció megvalósítható a spektrum transzformálásával vagy a spektrumból számolt (Bark) szűrőszávok szélességének és középfrekvenciájának módosításával. A transzformációs függvény paramétereinek meghatározásához számos technika áll rendelkezésünkre, például a lineáris diszkrimináns módszer (LD-VTLN), amely az osztályon belüli és osztályok közötti kovariancia hányadosát minimalizálja. Ezek az optimalizációs technikák azonban off-line módszerek, működésük feltétele, hogy az összes akusztikus adat előre rendelkezésre álljon. A beszélőnormalizáció előnyeinek kihasználásához az on-line rendszereknek más alternatív módszert kell választaniuk. Az általunk alkalmazott on-line RT-VTLN rendszer egy mesterséges idegháló segítségével becsülte meg a normalizációs paramétereket. A módszer eredményei azonban elmaradnak az off-line LD-VTLN módszer értékei mögött.

Az on-line RT-VTLN módszer hatékonyságának javítása érdekében az egyszerű statikus kombinációs stratégiák közül a „Prod”, „Sum”, „Max”, „Min” és a „Majority Voting”, valamint az aditív kombinációk közül az „AHP”, „Bagging” és a „Boosting” algoritmust alkalmaztuk. A kísérletek eredményeit a 3. táblázat foglalja össze.



3. táblázat. A kombinációs módszerek felismerési pontossága százalékban. A sávokat hármascsoportokba szerveztük, a csoportok felelnek meg a kombinációs módszernek, míg a csoporton belüli sávok a rendelkezésre álló adatbázist jelentik.

Minden kombinációs módszert 3 adatbázison értékeltünk ki, ezek a következők: „All”, „RT-VTLN” és a „Concat” adatbázisok, ez utóbbi a transzformált jellemzők mellett tartalmazza az eredeti jellemző-komponenseket is. A kombináció hatása függ az adatbázis komplexitásától. Az „All” adatbázison mindegyik kombinációs stratégia jobb eredményt ért el, mint az eredeti osztályozó. A transzformált jellemzőket tartalmazó adatbázisok esetében azonban az egyszerű statikus kombinációs módszerek nem tudtak számottevő javulást hozni. A legjobb eredményt a Bagging és a Boosting algoritmus alkalmazásával sikerült elérnünk. Összehasonlítva ezeket az eredményeket a LD-VTLN referencia (92,55%) értékével, megállapíthatjuk, hogy megfelelően megválasztott kombinációs sémával az on-line RT-VTLN (Boosting a Concat adatbázison, 92,67%) képes felülmúlni az off-line LD-VTLN

módszer eredményeit is.

Az artikulációs csatorna normalizáció kombinációs módszerekkel történő javítását [25] -ban publikáltuk.

Szófaji egyértelműsítés

A szófaji egyértelműsítés (POS tagging) során egy szöveg szavaihoz rendelünk szófaji címkéket, figyelembe véve a szó szótárban feltüntetett szófaját és a szó szöveggörnyezetét (azaz az adott szintagmában, mondatban, bekezdésben betöltött szerepét) is.

A szófaji egyértelműsítő módszerek néhány kombinációjának, a szavazásra épülő stratégiák és a többszintű döntési módszerek (például a „Stacking”) mélyreható elemzése megtalálható a [15]-ben. Magyar nyelvű alkalmazásokban egy másik, kaszkád felépítésű megoldás is kivizsgálásra került [19]. Az eredmények szerint [24] a kombinációs technikák segítségével a szófaji egyértelműsítő algoritmusok teljesítménye majdnem minden esetben javítható.

Azonban különböző adatbázisokon különböző kombinációs módszerek bizonyultak jobbnak, és nem tudunk következtetést levonni arra vonatkozóan, hogy mikor melyik kombinációs technikát használjuk. Robosztusabb kombinációs rendszerek építése érdekében meg kell vizsgálnunk, hogy a fejlettebb adaptív technikák, például a „Boosting”, hogyan alkalmazhatóak a szófaji egyértelműsítő alkalmazásokban.

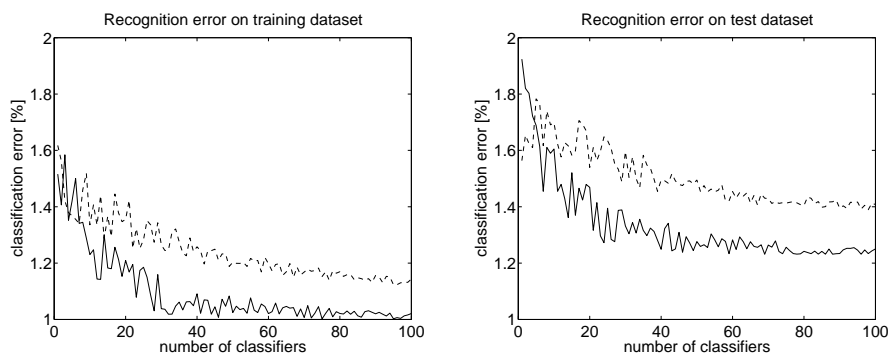
A Boosting algoritmus a független minták súlyozásán alapul, ahol a súlyokat az előző iterációk során konstruált osztályozók hibái határozzák meg. Minden iterációs lépésben olyan osztályozót próbál építeni, amely jól működik a nagyobb súlyú tanító példákon, míg a kisebb súlyú példák kisebb fontossággal bírnak a tanítási folyamatban.

A legtöbb algoritmus, mint például a TBL, nem képes az adatelemek, vagyis a szavak súlyozását kezelni. Ilyen esetekben a Boosting a tanító adatbázis elemének súlya, mint megfelelő valószínűségi eloszlás alapján az adatbázis újramintavételezett (Bootstrap) változatait fogja felhasználni az egyes osztályozók tanításához.

Ez a technika szimulálja a súlyozott elemek kezelését, azonban nem alkalmazható környezetfüggő feladatok esetén, mint például a szófaji egyértelműsítés. A természetes nyelvi feldolgozás során a korpusz szavai nem egy adatbázis független elemei, környezetük és pozíciójuk a mondatban mind befolyásolhatják jelentésüket.

A disszertációban bemutatunk egy általánosan használható megoldást, amely ilyen esetekben is alkalmazható. Tekintsük az osztályozót egy fekete doboznak, amely elemekhez rendel osztálycímkéket. Minden elem egy elemsorozat része, ahol az elemsorozatokat az osztályozó egymástól függetlenül (környezetfüggetlen módon) kezeli, míg az elemek jelentése függhet sorozatbeli környezetüktől is. Környezetfüggetlen alkalmazások esetén az elemsorozat csak egy elemet tartalmaznak, míg a szófaji egyértelműsítés során a sorozatok a mondatoknak vagy részmondatoknak feleltethetőek meg.

A Boosting algoritmus általánosított változata az elemek helyett az elemsorozatokhoz rendel súlyokat, és a bootstrap mintavételezés során teljes elemsorozatokat választ az eredeti adatbázisból. Az elemsorozatok osztályozási hibája kiszámítható elemeinek hibáiból, a publikált megoldásban az elemek hibáinak számtani összegét használtuk, és a kombináció hibáját a rosszul osztályozott elemek (súlyozott) relatív száma alapján számítottuk ki.



2. ábra. A Bagging (szaggatott vonal) és a Boosting (folytonos vonal) algoritmus osztályozási hibája a tanító és a teszt adatbázison.

A szófaji egyértelműsítés kombinációs módszerekkel való javításának vizsgálatát [22] -ben publikáltuk.

Szintaktikai Elemzés

A szintaktikai elemzés feladata az adott szöveg szintaktikailag összetartozó szósorozatainak azonosítása. Számos természetes nyelvi technológiai alkalmazásban játszik kulcsszerepet, mint pl. az információki-nyerés, a névelem-felismerés és több szövegbányászati alkalmazás.

A disszertációban magyar nyelvű szövegek főnévi szerkezeteinek elemzésével foglalkoztunk, és megvizsgáltuk a PGS algoritmus által generált szintaktikai elemzőket. Az elemző hatékonyságának növelése érdekében megvizsgáltuk, hogy a kombinációs stratégiák milyen hatással lehetnek a rendszer működésére.

Mivel a PGS algoritmus által generált elemző képes bizonyossági információt szolgáltatni, ezért számos kombinációs módszer alkalmazható. A szintaktikai elemző előállítja az adott mondat összes lehetséges szintaxis-fáját, ez a lehetséges osztályok halmaza. A kombináció során mindegyik elemző ugyanazon szintaxis-fához, vagyis osztályhoz tartozó kimeneteit összegezzük az adott kombinációs módszernek megfelelő módon, és azt a fát választjuk, amely a legjobb eredményt érte el. Ezt a stratégiát használva az egyszerű statikus kombinációk, mint a „Prod”, „Sum”, „Max”, „Min” és „Borda Count” sémák, könnyen integrálhatóak az elemző rendszerbe. Az adaptívan tanítható kombinációk esetében, mint a „Bagging” és a „Boosting”, a kombinációs algoritmus minden iterációs lépésben olyan elemzőket tanít, amelyek kompenzálják az előző lépések során felépített elemzők hibáját. Amennyiben a szöveg mondatait tekintjük független mintáknak, a Boosting algoritmus képes a PGS tanulót a problémás mondatok megoldására kényszeríteni, így a végső elemzés pontosságát javítani.

A kísérletekhez a tanító és a teszt adatbázisokat a Szeged Corpus üzleti hírei közül választott mondatok konvertálásával állítottuk elő. A kísérletek során 50 elemzőt generáltunk a PGS algoritmus különböző tanítóadatbázisokon való futtatásával, ezeket az adatbázisokat az eredeti adatbázisból 4000 mondat véletlenszerű kiválasztásával építettük fel.

A „Max”, „Min” és a „Prod” kombinációs sémák többé-kevésbé ugyanazt az eredményt érték el, használatukkal nem tudtuk a PGS algoritmussal generált nyelvtani elemzés teljesítményét növelni. A „Borda Count” és a „Sum” sémák már ki tudták használni a kombinációk előnyeit, alkalmazásukkal az $F_{\beta=1} = 82$ eredményt kaptuk a teszt adatbázison. A legjobb kombinációs teljesítményt a Boosting

Parser	$F_{\beta=1}$
Standalone parser	78.5
Prod Rule	79.4
Max Rule	77.5
Min Rule	78.7
Sum Rule	82.4
Borda Count	81.9
Bagging	83.6
Boosting	86.2

4. táblázat. A PGS elemzők kombinációinak eredményei

algoritmussal sikerült elérnünk az $F_{\beta=1} = 86$ eredménnyel, ami a kombáció nélküli PGS elemző pontosságával összehasonlítva hozzávetőlegesen 10% teljesítménynövekedést jelent.

A szintaktikus elemzés kombinációs algoritmusokkal való javítását [18] -ban publikáltuk.

Összegzés

A lineáris kombinációs technikák, különösen a Boosting algoritmus különböző változatai rövid idő alatt igen nagy népszerűsége tettek szert a gépi tanulási problémákban. Elméleti eredményei ellenére azonban a Boosting algoritmus a gyakorlati alkalmazásokban nem bizonyult mindig hatékonynak. Olyan esetekben, amikor nem áll rendelkezésünkre a több száz vagy akár több ezer osztályozó futtatásához szükséges erőforrás, egyszerűbb kombinációs megoldást kell keresnünk. A szerző által kidolgozott, analitikus hierarchikus folyamaton alapuló módszer egy hatékony kombinációs alternatívát biztosít az ilyen alkalmazások számára, amely a kísérletek eredményei is igazolnak.

A disszertációban megmutattuk, hogy a beszédtechnológiai alkalmazásokban a kombinációs stratégiák sikeresen alkalmazhatóak a fonémafelismerés javítására. A kutatócsoport által kifejlesztett OASIS beszédfelismerő rendszer segítségével a fonéma-osztályozók kombinációs technikáinak tanulmányozásához kísérleteket végeztünk. A fonémafelismerés fontossága miatt egy másik hatékony módszert, az artikulációs csatorna normalizációt és ennek kombinációs technikákkal való javítását is megvizsgáltuk.

A természetes nyelvi feldolgozás más alkalmazásai, mint a szófaji egyértelműsítés és a főnévi struktúrák elemzése, szintén érzékenyek az alkalmazott gépi tanulási módszerek hatékonyságára. A szakirodalomban számos egyszerű kombinációs stratégiát kidolgoztak ezekhez a rendszerekhez, azonban a probléma környezetfüggő tulajdonsága miatt a fejlebb Boosting algoritmusok nem használhatóak közvetlenül. A kidolgozott környezetfüggő Boosting algoritmus, a kísérletek szerint képes a vizsgált magyar nyelvű szófaji egyértelműsítési rendszerek és nyelvtani elemzők teljesítményének hatékony javítására.

1. Az eredmények tézisszerű összefoglalása

Az alábbiakban öt tézispontba rendezve összegezzük a Szerző kutatási eredményeit. A kutatásokból származó publikációkat, valamint azok tartalmának az egyes tézispontokhoz való viszonyát az 5. táblázat tekinti át.

- I.) A Szerző kidolgozott egy újszerű, az AHP módszerre épülő lineáris kombinációs stratégiát, amely képes akár kis számú osztályozó használásával is hatékonyan növelni az osztályozási pontosságot. A Szerző összevetette módszerét más, hasonló célú ismert módszerekkel, és megmutatta, hogy a javasolt technika az esetek többségében jobb osztályozási pontosságot eredményez, mint a hagyományos stratégiák.
- II.) A Szerző megtervezte és implementálta az OASIS beszédfelismerő keretrendszer alapvető részeit. Az integrált kombinációs modul felhasználva kiértékelte számos kombinációs technika hatékonyságát. A kísérletek igazolták, hogy többféle osztályozó-algoritmus kombinálásával a fonémafelismerés pontossága számottevő mértékben javítható.
- III.) A Szerző megvizsgálta a kombinációs stratégiák hatását az artikulációs csatorna hossznormalizáció során alkalmazott módszereken. Az eredmények azt mutatták, hogy az on-line módszerek megfelelő kombinációjával a rendszer képes felülmúlni akár az off-line módszerek hatékonyságát is. A javasolt módszerek beépítésre kerültek a "Beszédmester" beszédjavítás-terápiai és olvasásfejlesztő rendszerbe.
- IV.) A Szerző kidolgozta az Adaboost algoritmus egy általánosított, környezetfüggő változatát. A módszert magyar nyelvű szófaji egyértelműsítési problémán kiértékelve azt találta, hogy szignifikánsan képes javítani az osztályozási pontosságot a korábban használt kombinálási módszerekhez képest.
- V.) A Szerző megvizsgálta a főnévi csoportok elemzésére alkalmas P algoritmus kombinációs lehetőségeit. A kombinációs algoritmusokat a szintaktikai elemzés speciális igényeihez igazította. A kísérletek eredménye szerint a javasolt kombinációs stratégiák jelentősen javítottak a fáminták felismerési pontosságán.

	[10]	[11]	[25]	[22]	[18]
I.	•				
II.		•	•		
III.			•		
IV.				•	•
V.					•

5. táblázat.

Hivatkozások

- [1] Alexin, Z., Zvada, Sz., Gyimóthy, T., Application of AGLEARN on Hungarian Part-of-Speech Tagging, Second Workshop on Attribute Grammars and their Applications, WAGA'99, pp. 133-152, 1999.
- [2] Bishop C. M., Neural Networks for Pattern Recognition, Clarendon Press, 1995.
- [3] Brill, E., Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging, Computational Linguistics, Vol. 21, No. 4, pp. 543-565, 1995.
- [4] Breiman, L., Bagging Predictors, Machine Learning, Vol. 24, No. 2, pp. 123-140, 1996.
- [5] Breiman, L., Arcing classifiers, The Annals of Statistics, Vol. 26, No. 3, pp. 801-849, 1998.
- [6] Claes, T., Dologlou, I., Bosch, L., Compernelle, D., A Novel Feature Transformation for Vocal Tract Length Normalization in Automatic Speech Recognition, IEEE Trans. on Speech and Audio Processing, Vol. 6, pp. 549-557, 1998.
- [7] T.G. Dietterich, *Machine-Learning Research: Four Current Directions*, The AI Magazine, Vol. 18, No. 4, pp. 97-136, 1998.
- [8] Duda, R. O., Hart, P. E. and Stork, D. G., Pattern Classification, Wiley and Sons, 2001.
- [9] Eide, E., Gish, H., A Parametric Approach to Vocal Tract Length Normalization, ICASSP, pp. 1039-1042, 1997.
- [10] Felföldi, L., Kocsor, A., AHP-based Classifier Combination, Proceedings of PRIS-2004, pp. 45-58, Porto, 2004.
- [11] Felföldi, L., Kocsor, A., Tóth, L.: Classifier Combination in Speech Recognition, Per. Pol. Elec. Eng., Vol. 47, No. 1-2, pp. 125-140, 2003.
- [12] Freund, Y., Shapire, R., A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, Vol. 55, pp. 119-139, 1997.
- [13] Friedman, T. H. J., Tibshirani, R., Additive logistic regression, a statistical view of boosting, The Annals of Statistics, Vol. 28, No. 2, pp. 337-374, 2000.
- [14] Gosztolya, G. and Kocsor, A., Tóth, L., Felföldi, L., Various Robust Search Methods in a Hungarian Speech Recognition System, Acta Cybernetica, Vol. 16., pp. 229-240, 2003.
- [15] van Halteren, H., Zavrel, J., Daelemans, W., Improving data driven wordclass tagging by system combination, Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics, pp. 491-497, 1998.
- [16] Hócza, A., Alexin, Z., Csendes, D., Csirik, J., Gyimóthy, T., Application of ILP methods in different natural language processing phases for information extraction from Hungarian texts, Proceedings of the Kalmár Workshop on Logic and Computer Science, pp. 107-116, 2003.

- [17] Hócza, A. Noun Phrase Recognition with Tree Patterns, in Proceedings of the Acta Cybernetica, Szeged, Hungary, 2004.
- [18] Hócza, A., Felföldi, L., Kocsor, A.: Learning Syntactic Patterns Using Boosting and Other Classifier Combination Schemas, Proceedings of the 8th International Conference on Text, Speech and Dialogue, TSD 2005, LNAI, 3658, pp. 69-76, Springer Verlag, 2005.
- [19] Horváth, T., Alexin, Z., Gyimóthy, T., Wrobel, S., Application of Different Learning Methods to Hungarian Part-of-Speech Tagging, Proceedings of ILP99, LNAI, Vol. 1634, pp. 128-139, 1999.
- [20] Jain, A. K., Duin, R., Mao, J., Statistical Pattern Recognition: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 4-37, 2000.
- [21] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J., *On Combining Classifiers*, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20. No. 3, March 1998.
- [22] Kuba A. Jr., Felföldi L., Kocsor, A., POS tagger combinations on Hungarian text, Proceedings of the 2nd International Joint Conference on Natural Language Processing, IJCNLP, pp. 191-196, 2005.
- [24] Kuba, A., Hócza, A., Csirik, J. POS Tagging of Hungarian with Combined Statistical and Rule-Based Methods, Proceedings of the 7th International Conference of Text, Speech and Dialogue, pp. 113-121, 2004.
- [24] Kuba, A., Hócza, A., Csirik, J. POS Tagging of Hungarian with Combined Statistical and Rule-Based Methods, Proceedings of the 7th International Conference of Text, Speech and Dialogue, pp. 113-121, 2004.
- [25] Paczolay, D., Felföldi, L., Kocsor, A., Classifier Combination Schemes In Speech Impediment Therapy Systems, Acta Cybernetica, Vol. 17. No. 2, pp. 385-399, 2005.
- [26] Roli, F., Fumera, G., Analysis of Linear and Order Statistics Combiners for Fusion of Imbalanced Classifiers, 3rd Int. Workshop on Multiple Classifier Systems (MCS 2002), June, 2002.
- [27] Tóth, L., Kocsor, A., Gosztolya, G., Telephone Speech Recognition via the Combination of Knowledge Sources in a Segmental Speech Model, Acta Cybernetica, Vol. 16, pp. 643-657, 2004.
- [28] Tóth, L., Kocsor, A., Kovács, K., 2000. A Discriminative Segmental Speech Model and its Application to Hungarian Number Recognition, In: Sojka, P. et al. (eds.), Proceedings of Int. Conf. on Text, Speech and Dialogue TSD'2000, Lecture Notes in Artificial Intelligence Vol. 1902, pp. 307-313, Springer, 2000.
- [29] Valiant, L. G., A theory of the learnable, Commun. ACM, Vol. 27, No. 11, pp. 1134-1142, 1984.
- [30] Vapnik, V. N., Statistical Learning Theory, John Wiley & Sons Inc., 1998.
- [31] Wegmann, S., McAllaster, D. , Orloff, J., Peskin B., Speaker Normalization on Coversational Telephone Speech, ICASSP, Vol. 1, pp. 339-341, 1996.