A 2005-ös KDD kupa feladatának megoldása a Fürkész algoritmussal

KARDKOVÁCS ZSOLT, TIKK DOMONKOS, BÁNSÁGHI ZOLTÁN

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{kardkovacs,tikk}@tmit.bme.hu, empzoooli@gmail.com

Lektorált

Kulcsszavak: KDD kupa, internetes keresőkifejezések osztályozása, taxonómiák leképezése egymásra, szövegosztályozás

A 2005-ös ACM KDD kupa versenyfeladata internetes keresőkifejezések kategorizálása volt. Jelen tanulmányban ismertetjük a problémára adott megoldásunkat, amellyel a pontossági és kreativitási versenyben is második helyezést értünk el. A megközelítésünk túlmutat a konkrét feladat megoldásán: általános eszközt nyújt olyan rosszul specifikált osztályozási feladatokra, ahol nem áll közvetlenül rendelkezésre elegendő tanulóadat. Az algoritmus, amely az Internetet használja a szükséges tanulóadatok forrásaként, három részből áll: 1. probléma-specifikus adatszűrő; 2. webrobot konfigurálása az adatok szüretelésére; 3. hatékony osztályozó algoritmus alkalmazása. A módszerünkben kiemelt fontosságú a probléma megoldása során kifejlesztett általános algoritmusunk, amely képes különböző kategóriarendszereket egymásra leképezni.

1. A feladat

2005-ben kilencedik alkalommal írta ki az ACM szervezet KDD[1] szakcsoportja a KDD kupát[2]. A verseny az adat- és szövegbányászattal, valamint a gépi tanulással foglalkozó szakemberek (mind akadémiai, mind ipari területről) legrangosabb megmérettetése, amelyen évről évre egyre több kutatócsoport vesz részt a világ minden tájáról. Versenyfeladatként általában olyan problémát tűznek ki, amely a szakterület legaktuálisabb megoldatlan kérdéseit érinti. Az évek során a KDD kupákon számos nagy kihívást jelentő gyakorlati probléma lett kitűzve, melyek megoldásai hozzájárultak a tudományterület növekvő sikereihez.

A 2005-ös feladat az internetes keresések értelmének (kontextusának) meghatározásához kapcsolódott, ami szintén egy nehéz gyakorlati probléma. A legtöbb internetes keresés csak néhány szóból áll, azaz nagyon kevés információt tartalmaz a felhasználó keresési igényére vonatkozóan. Ha ez utóbbi rendelkezésre állna, akkor a keresőszolgáltatások hatékonysága nagymértékben javítható lenne.

A feladat megoldásának egyik számítástudományi megközelítése az, hogy megbecsüljük a keresőkifejezésnek egy adott taxonómia kategóriáihoz való hozzátartozását. Például a *jaguar* szóhoz egyaránt hozzárendelhetjük a *zoológia* és az *autó* kategóriákat is. Ezzel a módszerrel a keresési igény meghatározásának kérdése a legvalószínűbb kategóriák kiválasztására redukálódik – egy tetszőleges, de adott taxonómia alapján.

A KDD kupa 2005-ös feladata egy konkrét taxonómia feletti osztályozási probléma volt: 800.000 internetes keresőkifejezést kellett tartalmuk szerint 67 előre megadott kategóriába besorolni[3]. Minden keresőkifejezéshez legfeljebb 5 kategóriát kellett rendelni rangsorolás nélkül.

A probléma megoldását külön nehezítette a rendkívül nagy méretű és rossz minőségű adattömeg, valamint hogy egyáltalán nem állt rendelkezésre tanulóadat, ami az osztályozás típusú gépi tanulási problémák esetén feltétlenül szükséges. További nehézséget jelentett, hogy a szervezők menet közben megváltoztatták a kategóriarendszert. A feladat nehézségére jellemző, hogy a több mint 140 regisztrált résztvevő közül csak 32 csapat adott be megoldást.

A cikkben az alábbi terminológiát fogjuk használni:

• **céltaxonómia:**

A szervezők által meghatározott 67 kategória halmaza, amelybe be kell sorolni a keresőkifejezéseket. Itt azért használjuk a „taxonómia” kifejezést, mert a megoldásunk során a kategóriákat kétszintű hierarchiába rendeztük, úgy, hogy egyes összetartozó kategóriákat egy közös felső szintű kategória alá soroltunk. Így a *Computer, Entertainment, Living* stb. lettek az új csúcskategóriák és az eredeti *Computer/Hardware, Computer/Software* stb. kategóriák lettek a levélszintű kategóriák.

• **célkategória:**

A céltaxonómia egy levélszintű kategóriája.

• **keresőkifejezés:**

Általában a 800.000-es keresőkifejezés egy elemét értjük alatta, ha másképp nem állítjuk; a szövegosztályozási paradigma alapján esetenként használjuk erre a *dokumentum* terminológiát is.

• **szótövezett keresőkifejezés:**

Ugyanaz mint előbb, de a keresőkifejezés szavai szótövezve vannak.

1 Association of Computing Machinery, http://www.acm.org; Knowledge Discovery and Data Mining
2 http://www.acm.org/sigs/sigkdd/kdd2005/kddcup.html
3 A cikkben a rögzített kategóriarendszerbe történő besorolás feladatára egyaránt használjuk a besorolás, osztályozás, kategorizálás megnevezéseket.

• **forrástaxonómia:**
Internetes keresőszolgáltatás által nyújtott kategóriarendszer, amely segíti a felhasználókat dokumentumok keresésében és témák közti navigálásban.

2. Bevezetés

A dokumentumok rögzített kategóriarendszerbe való besorolását szövegosztályozásnak nevezik. A KDD kupa 2005-ös versenykiírása is nyilvánvalóan ebbe a feladatkörbe tartozik. A szövegosztályozás tipikus felügyelt gépi tanulással feladat; a tanuló algoritmus ismert mintaadatok segítségével „megtanulja" a kategóriarendszer sajátosságait, majd ezután ismeretlen, korábban nem látott dokumentumokat a megtanult jellemzők figyelembevételével osztályoz.

A kupa kiírása azonban tartalmazott néhány olyan lényeges eltérést a standard feladattól, amely miatt a feladatot nem lehetett egyszerűen, valamely ismert algoritmus alkalmazásával megoldani:

1. A dokumentumok (itt: keresőkifejezések) nagyon rövidek voltak, a 90%-uk 5 szónál rövidebb.
2. A korpusz nagyon zajos volt, legalább 30%-ában rossz karakterkódolású nem-angol szövegeket, illetve teljesen értelmetlen szövegeket tartalmazott.
3. A leglényegesebb eltérés az volt, hogy nem állt rendelkezésre tanulóadat: a mellékelt 111 mintapélda csak a céltaxonómia szemantikájának illusztrálását szolgálta.

Ebből következően a mintapéldákat csak validálási célra lehetett korlátozott mértékben felhasználni. Ennek ellenére úgy véltük, hogy a feladatot csak felügyelt tanulással érdemes megközelíteni, mivel

- A keresőkifejezések nagy számú tulajdonnevet tartalmaztak, amelyek hatékony figyelembevétele csak a szótárak vagy tanulóadatok segítségével lehetséges.
- Ha létezett volna a céltaxonómiához hasonló forrástaxonómia megfelelő dokumentumokkal, azt csak akkor lehetett volna felhasználni, ha valamilyen leképezés rendelkezésre állt volna a forrástaxonómia és a céltaxonómia kategóriái között. Mivel azonban a célkategóriák szemantikája nem volt ismert, ezzel bármely ilyen leképezés érvényessége megkérdőjelezhetővé vált.

Összegezve úgy véltük, hogy a megoldást a keresőkifejezések szemantikailag indokolt kibővítése és tanulóadatokként való felhasználása jelentheti, amely lépés után a felügyelt tanulás paradigmája már alkalmazható.

A módszert „Fürkész algoritmusnak" neveztük el, amely a következő lépésekből áll *(1. ábra)*:

1. *Forrás megtalálása:* Határozzuk meg a céltaxonómia kategóriáinak szemantikáját a keresőkifejezések tartalmából kinyert kiindulási szótár létrehozásával, majd adjunk meg ezen szemantika alapján érvényes leképezést internetes keresőszolgáltatások forrástaxonómiája, valamint a céltaxonómia között.
2. *Dokumentumok szótövezése:* Hajtsunk végre szótövezést az összes keresőkifejezésen, így kapjuk a szótövezett keresőkifejezéseket.
3. *Keresés az Interneten:* A cél- és forrástaxonómiák közti kapcsolat alapján a szótövezett keresőkifejezéseket küldjük el a forrástaxonómiákhoz kapcsolódó keresőszolgáltatások felé tanulóadatok gyűjtése céljából.
4. *Eredmények feldolgozása:* Dolgozzuk fel a 3. pontban kapott eredményeket. Az egyes keresőkifejezésekhez kapott eredményoldalakból és a taxonómiák közti leképezésből meghatározzuk a keresőkifejezés kategóriáját. Az eredményoldalak tartalmát tanulóadatként az adott kategóriához rendeljük.
5. *Osztályozó betanítása:* A 4. pontban kapott tanulóadatokkal tanítsunk be egy tetszőleges szövegosztályozót. Munkánk során a HITEC hierarchikus osztályozót használtuk [1,6,7], részleteket lásd a 3.4. szakaszban.
6. Futtassuk az osztályozót a keresőkifejezésekre. Rangsoroljuk az Internet keresés és az osztályozás eredményeit, és határozzuk meg a legjobb 5 kategóriát minden keresőkifejezésre.

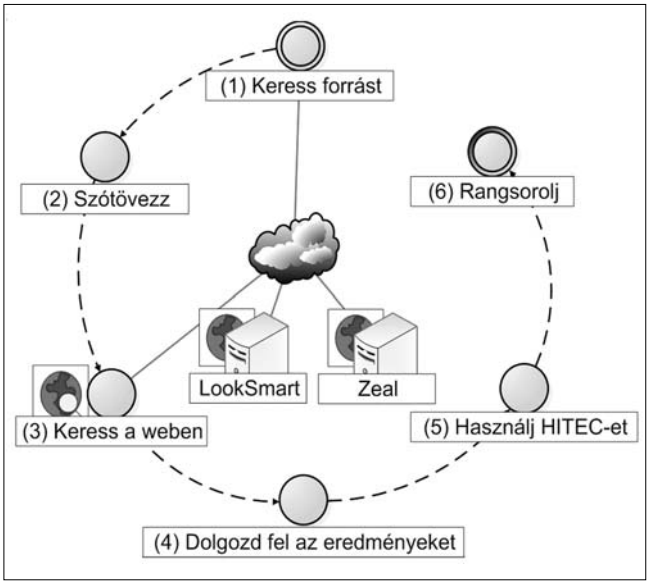
A következő szakaszokban a fenti lépéseket részletesen bemutatjuk.

3. A Fürkész algoritmus

3.1. Az Internet mint tudásbázis

A Fürkész algoritmus lényege, hogy az Internet-et használja tudásbázisként azáltal, hogy különböző keresőszolgáltatásokat használ fel a keresőkifejezések szemantikájának meghatározására. Tehát az algoritmust egy metakeresőnek is tekinthetjük.

1. ábra
A Fürkész algoritmus lépései



A KDD kupa feladatának megoldása során a LookSmart (http://search.looksmart.com) és a Zeal (http://www.zeal.com) [4] keresőszolgáltatásokat használtuk. A választásunk azért esett erre a kettőre, mivel némi párhuzam felfedezhető a céltaxonómia és a keresőmotorok forrástaxonómiái közt, illetve a két forrástaxonómia is hasonló, ezért megoldható feladat volt köztük leképezést meghatározni. A keresőszolgáltatások által adott eredményeket lokálisan dolgoztuk fel.

Mindazonáltal a LookSmart és Zeal teljes forrástaxonómiáinak újraépítése nem tűnt célszerű feladatnak, mivel ezek nagyon nagyméretű és szövevényes rendszerek. Ezek a taxonómiák ugyanis lehetőleg teljesen le kívánják fedni a felhasználók érdeklődési körét, hiszen a szolgáltatások fő célja, hogy hatékonyan segítse a meghatározott céllal kereső felhasználót internetes oldalak és regionális információk megtalálásában. Ezért a Fürkész algoritmussal csak azt a részgráfját térképeztük fel a forrástaxonómiáknak, amelyhez a keresőkifejezések által meghatározott kategóriák tartoztak. Ezt a szolgáltatások publikus keresési felületének alkalmazásával tettük meg, leszüretelve az eredményoldalak egy részét a későbbiekben történő feldolgozásra.

Szintén jól felhasználható forrás volt a fenti szolgáltatások esetén, hogy minden kategóriához egy rövid meghatározás is tartozott. Ez alapján hoztuk létre a célkategóriák szemantikáját leíró kiinduló szótárat.

3.2. Kategóriák szemantikája

Mivel meg voltunk győződve arról, hogy a KDD kupa feladatát felügyelt tanulással célszerű megoldani, ezért a tanulóadatok begyűjtése elsődleges céllá vált. Ezt a következőképpen valósítottuk meg. Először minden keresőkifejezést a Porter-eljárással [2] szótöveztünk, majd ezeket elküldtük a keresőszolgáltatásoknak.

Az LZ eredményoldalak két fő részből álltak [5]: ha a keresésnek van eredménye, akkor az egyik rész a találat rövid kivonata *(kontextus)*, illetve ha a találat a keresőmotor szerkesztői által be lett sorolva a forrástaxonómia valamelyik kategóriájába *(forráskategória)*, akkor a találat másik része a kategóriának a neve.

Abban az esetben, ha vannak még jellemző kategóriák a keresőkifejezésre, akkor ezekből a legfontosabbakat egy külön szekcióban jelzik. Emellett a fontosabb kategóriákhoz a kategória tartalmát jellemző rövid leírás is adott. A forrástaxonómia szerkezetét, illetve az egyes forráskategóriák gyökérkategóriából való elérhetőségét gyerek-szülő relációk alapján térképeztük fel.

Ezzel a módszerrel megkaptuk a két forrástaxonómiának a feladat megoldása szempontjából releváns részstruktúráját, valamint néhány szavas szemantikai leírást a forráskategóriákra. A leírások összességét kiindulási forrásszótárnak *(BC)*, a kapcsolódó kategóriákat pedig *BC*-kategóriáknak nevezzük.

3.3. A taxonómiák közti leképezés meghatározása

A taxonómiák közti leképezés meghatározásának alapja a célkategóriákhoz tartózó maximális releváns szóhalmaz meghatározása.

Feltesszük, hogy a célkategória neve a lehető legjobban leírja az adott kategóriát. Ha ez fennáll, akkor a célkategória nevében szereplő szavak szinonimái szintén jól írják le a kategóriát. Ezt a lépést a szavak WordNet [6] szinonimáival való kibővítésével végeztük, és az eredményt a célkategória *szemantikus lezárt*-jának neveztük.

Legyen $W(0) = \bigcup_i w_i(0)$

a céltaxonómiát leíró kiinduló célszótár, ahol $w_i(0)$ jelöli az *i*-edik kategória szemantikus lezártját.

A $W(0)$ halmazból kiindulva kerestünk releváns *BC*-kategóriákat, vagyis ahol a *BC*-kategória szemantikai leírása és a kiinduló célszótár közös elemeket tartalmaz. Formálisan, legyen $C_i^0 \subseteq BC$ azon része a forrásszótárnak, amelyet a $w_i(0)$ célkategória szemantikus lezártja meghatároz, vagyis amely kategórialeírások metszete $w_i(0)$-lal nem üres. A Fürkész algoritmus a jól ismert TF-IDF mértéket számolja ki (pl. [3]) a C_i^0 leírások szavaira, amelyeknek relatív gyakorisága legalább ω legalább egy C_i^0-beli leírásban, és legfeljebb α számú C_i^0-beli leírásban fordul elő. (Ezzel a túl ritka és túl gyakori szavak kiszűrését tudjuk parametrikusan megvalósítani). Legyen A^0 azon szavak halmaza, amelyre ez a tulajdonság fennáll. Ekkor a következő rekurzív formulát alkalmaztuk:

$$w_i(n+1) = w_i(n) \cup \{a \mid a \in A^n \cap C_i^n\} \quad (n = 0, 1, \ldots) \quad (1)$$

Ezt a lépést nemcsak az eredeti, a céltaxonómiában levél szinten lévő 67 kategóriára végeztük, hanem a felső szintű kategóriákra is. Ekkor úgy jártunk el, hogy a gyerekkategóriák halmazainak unióját képeztük, és arra alkalmaztuk a fenti eljárást. Ez azért fontos lépés, mert az IDF tényező egyes gyakori szavakat – amelyek például a kategóriák egy csoportjára jellemzőek – kiszűrhet, de ekkor ezeket még felső szintű kategóriára jellemző szóként figyelembe vehetjük. A felső szintű célkategóriáknak a kategóriák meghatározásánál is szerepük van: ha egy dokumentum csak a *Computer* kategóriához tartozik, de egyik alkategóriájához sem, akkor mint egyéb lesz osztályozva *(Computer/Other)*.

A fenti (1) képlettel leírt rekurzív algoritmus a leírásban szereplő szavak végessége miatt nyilván terminál. Azok a forráskategóriák

$$C^- = BC \triangleleft \bigcup_i C_i^N$$

(itt $\triangleleft$ a halmazok közti különbségképzés jele),

melyhez nem rendeltünk célkategóriát, úgy kerülnek felhasználásra, hogy a forrástaxonómia relációi mentén a legközelebbi olyan szülőkategóriához rendeljük őket, amelynek már a céltaxonómiában van párja.

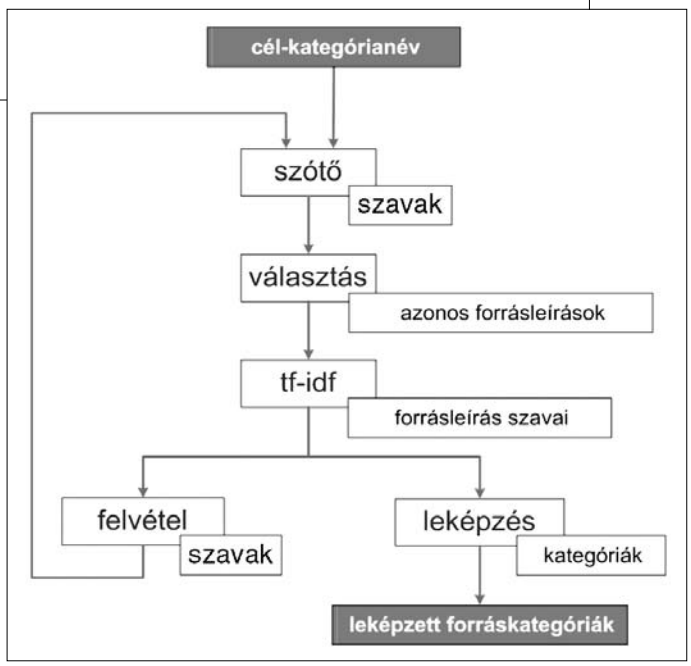
4 Együtt a kettőre LZ-ként hivatkozunk a továbbiakban.
5 A letöltéseket 2005 júliusában végeztük, azóta a Looksmart kereső honlapja és eredményoldalainak szerkezete megváltozott.
6 http://wordnet.princeton.edu/

Az ilyen $C^+ = BC \triangleleft C^-$ halmazbeli forráskategóriákat *megjelölt kategóriának* nevezzük. Vegyük észre, hogy az algoritmussal megkapjuk mind a célkategóriák szemantikus leírását, w_i^N-t, mind a cél- és forrástaxonómia közti leképezést; ezt minden *i* célkategóriára a C_i^N halmazban lévő forráskategóriák adják meg. Az algoritmus folyamatábrája és pszeudokódja rendre a *2. és 3. ábrákon* látható.

2.-3. ábra
A taxonómiák közti leképezést meghatározó algoritmus pszeudokódja és a leképezés meghatározásának lépései

bemenet: egy részfa struktúra és annak gyökere
kimenet: egy leképezés a forrás és célkategóriák között

```
function katLeképzés( Csomópont Gyökér, Gráf Fa ) begin
    let gyakorisági_küszöb := 0.05;
    let ismétlôdési_küszöb := 3;
    if ( Gyökér is levél )
        return nil;
    foreach S in Gyökér Fa-beli gyermekei
    begin
        let ÚjSzók[S] := { S morfológiai alaptöve és szinonimái };
        let KatJelöltek[S] := Üres;
        let Szók[S] := Üres;
    end
    while ( ÚjSzók != Szók )
    begin
        let Szók := ÚjSzók;
        foreach S in Gyökér Fa-beli gyermekei
        begin
            let ÚjKatJelölt[S] := KatJelöltek[S] + { forráskategóriák,
amelyek tartalmazzák Szók[S] egy elemét };
            let ÚjSzók[S] := { ÚjKatJelölt[S] leírásában szereplô azon
szavaknak morfológiai tövei, amelyre a tf > gyakorisági_küszöb };
        end
        let ÚjSzók := azon szavak halmazainak tömbje, amelyek legfeljebb
ismétlôdési_küszöb kategoriában fordulnak elô: idf < ismétlôdési_küszöb;
        let KatJelöltek := azon kategóriahalmazok tömbje, amelyre igaz,
hogy egy kulcs csak egy kategória alá lett besorolva;
    end
    return KatJelöltek;
end
```



Vegyük észre, hogy a 3. ábrán ismertetett algoritmus tetszőleges forrás- és céltaxonómia összekapcsolására alkalmas, ha a forrástaxonómia kategóriáinak van szemantikus leírása. Ez a feltétel könnyen teljesíthető, ha a forrástaxonómiához adottak tanulóadatok, ekkor ugyanis a szemantikus leírást a kategóriákhoz rendelt tanulóadatokban szereplő leggyakoribb szavak összességeként kaphatjuk. A kategóriáknak szóprofilja, vagy prototípus vektora szintén tekinthető szemantikus leírásnak.

3.4. Tanítás és osztályozás

Miután a leírt módon a kategóriákhoz tanulóadatokat gyűjtöttünk, a HITEC [1,6,7] osztályozót alkalmaztuk a keresőkifejezések osztályozására. Erre azért volt szükség, mert az összes keresőkifejezésnek csak mintegy feléhez sikerült a fenti módszerrel kategóriát rendelnünk. A HITEC neurális hálózat alapú osztályozó, amely a tanulóadatok alapján minden kategóriához egy prototípus vektort készít. Amikor ismeretlen dokumentumokat osztályoz, akkor a prototípus vektorokhoz való hasonlóság alapján határozza meg a taxonómiában lefelé haladva a dokumentum releváns kategóriáit. A HITEC-nek számos dokumentumfeldolgozást, tanulást és következtetést szabályozó paramétere van. Ez utóbbiak főleg a taxonómiában való keresés szélességét és mélységét befolyásolják.

További részletek a hivatkozásokban találhatók.

3.4.1. A tanulóadatok

Mivel a keresőkifejezések eleve nagyon kevés szót tartalmaztak, ezért a csak ezek tartalmából épített szótár mérete is kicsiny, és nem bír elegendő leíróképességgel az ismeretlen dokumentumok osztályozására. Ezt javítandó, 4 alternatív tanulóhalmazt alakítottunk ki, amelyeket az eredeti dokumentumoknak az LZ eredményoldalak tartalmával való bővítésével kaptunk meg. Az eredményoldalak szerkezetének leírását az Olvasó a 3.2. szakaszban találja.

Futás	d_1	d_2	w_t	w_d	$\|D\|$	Tanulóhalmaz
R1	2	0,7	5	3	888.565	K+T+C
R2	5	1,0	10	3	149.792	K+C
R3	5	1,0	10	3	809.165	K+T+C
R4	0	1,0	10	3	1.088.171	SK+T+C
R5	1	1,0	10	3	763.872	T+C

1. táblázat
A szótár mérete és a tanulóhalmazok közti összefüggés

Az első három alternatíva forráskategóriákhoz rendel tanulóadatokat az LZ eredményoldalak alapján. A tanulóadat maga a keresőkifejezés, vagy annak a lent leírt módon való valamilyen bővítése. Olyan forráskategóriákhoz rendeltünk ily módon tanulóadatokat, amelyek legalább egyszer előfordultak a keresőkifejezések eredményoldalán. A forrás- és célkategóriák közti kapcsolatot a korábban ismertetett leképezés (lásd a 3.3. szakaszt) segítségével hoztuk létre. A negyedik alternatív tanulóhalmazt közvetlenül a forráskategóriák alapján határozzuk meg, a keresőkifejezések használata nélkül.

- **K** – KERESŐKIFEJEZÉS:
  Ez a legegyszerűbb eset, amikor tanulóadatként magát a szótövezett keresőkifejezést alkalmaztuk. Ezt minden olyan forráskategóriához (és a leképezésen keresztül célkategóriához) hozzárendeltük, amely az adott kereséskor eredményként legalább egyszer előfordult.
- **SK** – SÚLYOZOTT KERESŐKIFEJEZÉS:
  Az előzőhöz hasonló, de ekkor a szótövezett keresőkifejezést annyiszor rendeltük a forráskategóriához, ahányszor az adott kereséskor eredményként előfordult. Vagyis ha a *q* keresőkifejezés eredményoldalán a *c* kategória kétszer fordult elő, akkor a *q* szövegét kétszer egymás után összefűzve rendeltük *c*-hez.
- **T** – TEXT:
  A szótövezett keresőkifejezés szövegét az alábbi módon bővítettük. Az eredményoldalak letöltése után kinyertük a szöveges ASCII információt a HTML oldalakból, és evvel kibővítettük a keresőkifejezést. A kategóriákhoz analóg módon rendeltük az így kapott tanulóadatot.
- **C** – KATEGÓRIALEÍRÁS:
  A keresőszolgáltatások által adott rövid szemantikai leírás szövegét rendeltük a forráskategóriákhoz. Vegyük észre, hogy ez független a keresőkifejezésektől. A leírás címből és általában egy mondatból áll, amelyeket külön mezőként kezelve különböző súllyal láttunk el (lásd 1. táblázat). Ezt a módszert csak megjelölt kategóriákra alkalmaztuk.

3.4.2. A jellemzők kiválasztása

A tanulás egyik legfontosabb része a megfelelő jellemzők – szövegosztályozás esetén az optimális szótárméret – meghatározása. A cél a lényegtelen, ritka szavak elhagyása és a megkülönböztető szavak megtartása közti optimális egyensúly elérése.

A HITEC két egyszerű paraméterrel rendelkezik a szótár méretének ($|D|$) szabályozására:

- $d_1 \in [0, \infty)$ a szavak korpuszban való minimális előfordulására vonatkozó alsó küszöbérték;
- $d_2 \in (0, 1]$ pedig a szavak korpuszon való teljes eloszlására vonatkozó felső küszöbérték.

Ezek a paraméterek a TF-IDF súlyozási sémával kapcsolatosak: d_1 és d_2 rendre a TF-re vonatkozó alsó, illetve az IDF-re vonatkozó felső korlátnak tekinthető.

Az adott feladat esetén, amikor K, T vagy C tanulóhalmazokat (és ezek kombinációit) alkalmaztuk, nagyon alacsony d_1 (2-5) és nagyon magas d_2 (~0,5) értékekkel számoltunk, mivel az egész korpusz csak kevés szót tartalmazott, és jellemzően azok eloszlása is ritka volt. A d_1 paraméter alacsony értékét az is indokolta, hogy ebben sok értékes, megkülönböztető jelleggel bíró szó csak 1-2-szer fordult elő. A C tanulóhalmaz használatakor két súlytényezőt alkalmaztunk: w_t-t a címre, és w_d-t a leírásra.

A legnagyobb tanulóhalmaz esetén (K+T+C), a teljes szótár szótövezés után 1080 ezer szót tartalmazott. Jellemző-kiválasztással ennek méretét 809 ezerre csökkentettük. A K+C tanulóhalmaz esetén csak 149 ezer szó volt a szótárban. Az *1. táblázatban* feltüntettük néhány jellemző-kiválasztási futás eredményét.

3.4.3. A tanulás paraméterei

Az iteratív tanulás során az iterációk számát 5-nek rögzítettük, ami korábbi nagy korpuszokon való osztályozási feladatoknál (pl. Reuters Corpus Volume 1) kapott kvázi-optimális érték. A tanulás során minden tanulóadatot felhasználtunk.

A HITEC-ben két fontos paraméter szolgál az osztályozási következtetés eredményhalmaza méretének szabályozására. A maximális varianciával ($v_{max} \in (0, 1]$) megadható, hogy a legnagyobb konfidenciaszintű csomóponthoz képest milyen arányú eltérést engedünk a taxonómia egy adott szintjén, amikor a további keresésnél figyelembe vett csomópontokat határozzuk meg.

Ha ezt az értéket alacsonyra állítjuk (kb. 0,5), akkor minden szinten több kategóriát választunk ki, és így a következtetés több szálon fut. A küszöbérték, θ segítségével a kiválasztáshoz szükséges minimális konfidenciaérték adható meg. Ha ez az érték alacsony (0,05~

0,15), akkor ismét több kategóriát kapunk eredményül. Ezzel a két paraméterrel tehát a felidézés és a pontosság duális mértékek közti egyensúlyt lehet beállítani; alacsony paraméterértékek esetén az előbbi nő, az utóbbi viszont csökken.

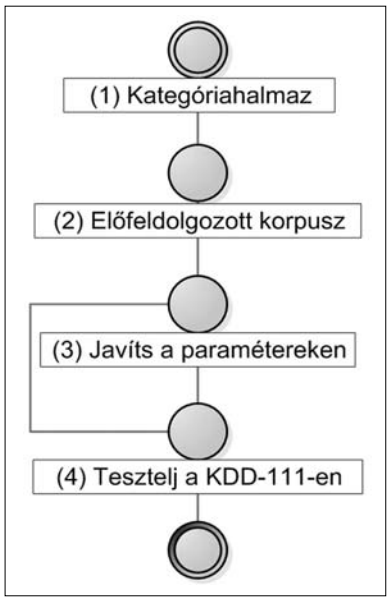
A KDD kupa feladatánál ezeket az értékeket alacsonyra állítottuk, hogy a lehető legtöbb keresőkifejezésre megkapjuk a szükséges öt eredménykategóriát: v_{max}= 0,5 és θ = 0,1.

A HITEC lehetővé teszi, hogy a szótárkészítéshez, és a tanuláshoz különböző tanulóhalmazokat használjunk. Ezt kihasználva a legtöbb futás során a tanulásnál figyelmen kívül hagytuk a K tanulóhalmazt, mivel azt tapasztaltuk, hogy jelenléte rontja a tanítást. Ez annak köszönhető, hogy ezek a dokumentumok túl rövidek és zajosak. Másrészt viszont K felhasználása a szótár létrehozásánál hasznos, mivel több fontos szó előfordulási értékét növeli –, például a K+T+C tanulóhalmaz alkalmazása esetén.

Egyes beállítások tanulási kapacitását – azaz, hogy a HITEC mennyire volt képes a tanulóadatokat megtanulni – a *2. táblázatban* ismertetjük. A *4. ábrán* az iterációk során elért tanulási hatékonyságot ábrázoljuk a HITEC belső kiértékelő függvényének segítségével. A bemutatott hatékonysági mértékeket a tanulóadatokon történt teszteléssel kaptuk meg.

3.4.4. Kiértékelés

A különböző jellemzőkiválasztási verziókat és tanulási beállításokat a megadott 111 mintaadaton teszteltük (lásd még az *5. ábrát)*. A kiértékelés során az LZ eredményoldalak közvetlen feldolgozásával kapott kategóriák hatékonyságát is vizsgáltuk. Itt hasonló módon – a taxonómiák közti leképezés alapján – jártunk el, mint ahogy a keresőkifejezésekhez tartozó célkategóriákat meghatároztuk. Az összehasonlítás során megállapítottuk, hogy a 111 mintaadaton a HITEC által adott következtetés lényegesen rosszabb eredményt adott, mint amit az LZ oldalak közvetlen feldolgozásával kaptunk, függetlenül a HITEC paramétereinek beállításától. Ez egyáltalán nem meglepő, hiszen a HITEC eredményeiben kétféle hiba kumulálódik: egyrészt a tanulóhalmaz szintén nem elhanyagolható mértékű hibája, másrészt a tanulás és az osztályozás hibája. Ráadásul a 111 mintapéldában számos olyan szó van, amely az egész korpuszban nem fordul elő egyszer sem (pl. *aldactone*), amire tehát az osztályozó nem tud értelmes következtetést adni.



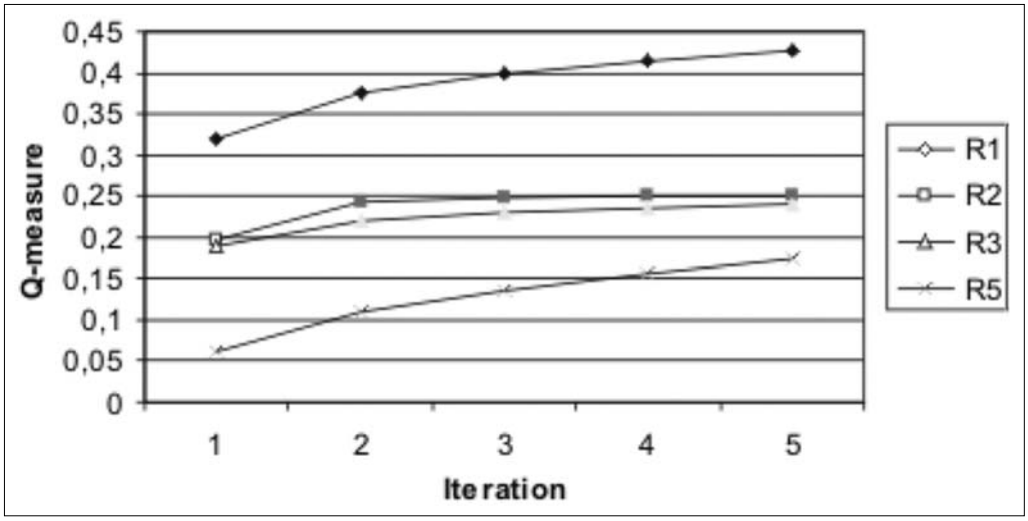
5. ábra

Kiértékelés a HITEC-kel

Futás	v_{max}	θ	F_1	Pontosság	Felidézés	Adatok
R1	0,5	0,1	0,655	0,586	0,741	T+C
R2	0,5	0,1	0,532	0,459	0,634	K+C
R3	0,5	0,1	0,517	0,446	0,616	T+C
R5	0,7	0,2	0,326	0,280	0,391	K+T+C

2. táblázat

Validációs eredmények egyes futásokra (jelölések az 1. táblázat alapján)



4. ábra

A tanulás hatékonysága a HITEC belső kiértékelő függvénye alapján

A tanulás hatékonyságát két tényezővel ellenőriztük. Egyrészt a 111 mintaadaton, másrészt a validációs értékek F_1 mértéke alapján. Ez alapján azt találtuk, hogy a legígéretesebb eredményt a (T+C) tanulóhalmaz adta a d_1= 2, v_{max}= 0,5 és θ = 0,1 értékek mellett. Idő hiányában nem volt lehetőségünk a HITEC optimális beállításának meghatározására. (Az R1 futás már a beadási határidő után ért véget, ezért az R2 futás eredményét adtuk be.)

4. A beadott eredmények

A beadott eredmények két forrásból származtak:

1. Azokra a keresőkifejezésekre, amelyekre az LZ keresések adtak forráskategóriát, eredményként a leképezés által megadott célkategória(ka)t adtuk be. Ezáltal mintegy 400 ezer keresőkifejezésre kaptunk eredményt.

2. A többire a HITEC betanított modelljének legnagyobb 5 konfidenciaértékű következtetését adtuk be. Ezzel további mintegy 320 ezer keresőkifejezésre kaptunk eredményt.

3. A maradék körülbelül 80 ezer keresőkifejezésre nem adtunk be eredményt. Ezek túlnyomórészt olyan keresőkifejezések voltak, amelyek nem tartalmaztak értelmes angol szavakat, vagy tulajdonneveket.

A beadott megoldásokat a szervezők mindössze 800, három szakértő által osztályozott adaton tesztelték. Az általunk beadott eredmények 0,340883 pontosságot és 0,34009 F_1-mértéket értek el. Az előbbi értékkel, illetve a Fürkész algoritmus ötletességével második helyezést értünk el a pontossági és a kreativitási versenyben.

Utólag megvizsgáltuk, hogy az algoritmusunk egyes komponensei milyen mértékben járultak hozzá az elért eredményhez. A 800 adatból 665-re volt válasz a tanulóadatokban (1. csoport). A maradék 135 keresőkifejezésre (2. csoport), amennyiben volt válasz, a HITEC által szolgáltatott eredményt adtuk be.

Akadt 800 között néhány olyan keresőkifejezés is, amire a HITEC sem adott megoldást, tehát a 3. csoportba tartoztak (lásd *3. táblázat* „Nincs címke” oszlopa). Az ezek nélkül számolt értékek a *4. táblázatban* találhatók, ahol nyilván csak a felidézés és az F_1 értékek változnak.

A HITEC válaszaira megnéztük, hogy a legjobb három futás (R1, R2, R3) milyen eredményt adott. Vizsgálatunk érdekes eredményt adott: a legjobb eredményt az R3 futás adta, míg a legrosszabbat az R1, ami éppen a tanulóadatokon mutatott viselkedés fordítottja (részleteket lásd a 3. és 4. táblázatban).

Ezt két okkal magyarázhatjuk. Egyrészt a tanulóadatokon való kedvező viselkedés azokon a keresőkifejezéseken való hatékony tanulást jelenti, amelyekre az 1. csoportból adtunk be eredményt, tehát az ezekre való tanulási képesség a kiértékelésünkben nem játszott szerepet. Másrészt a jobb validációs eredményű tanulásnál túltanulás lép fel, ezért az így betanított osztályozó általánosító képessége kisebb. A hatékonyság alacsony számértéke megtévesztő: ezek pont a legnehezebb, tanulóadatokkal nem rendelkező keresőkifejezésekre adott válaszok, tehát itt ennél lényegesen jobb eredmény nem is várható el.

5. A versenyen díjazott további módszerekről

5.1. Osztályozók kombinációja

Az első helyezést mindhárom kategóriában a hongkongi HKUST egyetem csapata érte el [5]. Megoldásukban egy többkomponensű osztályozót készítettek, melynek a vázlata a *6. ábrán* látható.

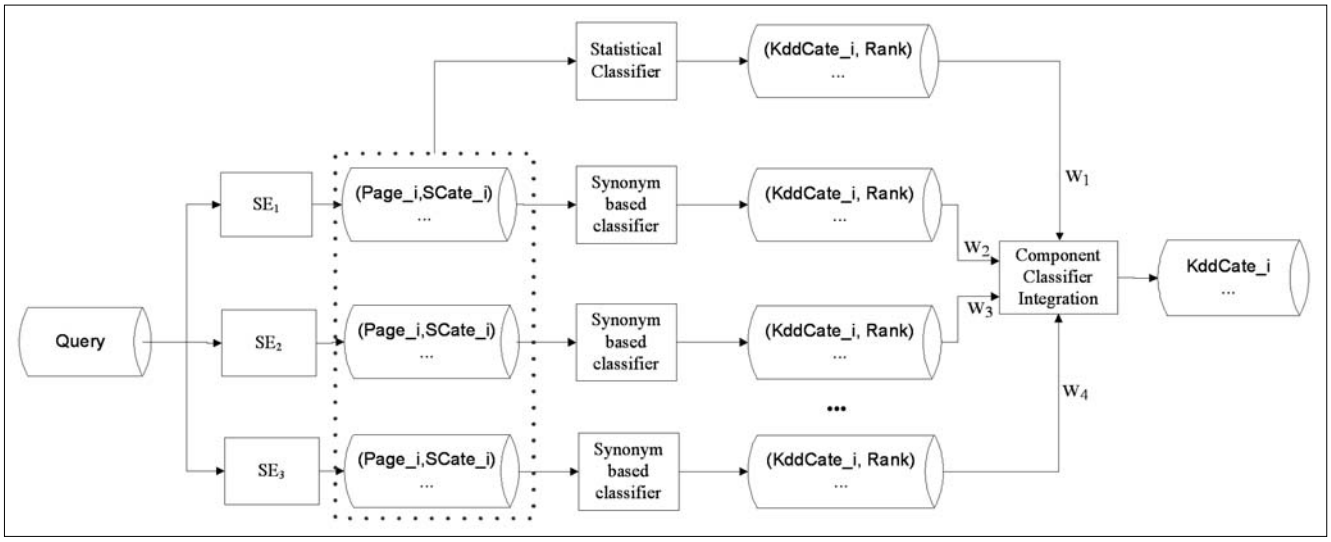
A keresőkifejezések és a céltaxonómia elemei közti hozzárendelés megvalósításának első lépése a keresőkifejezések internetes keresőmotorok (SE) segítségével való kibővítése. Három keresőmotort használtak fel erre a célra: a Looksmartot, a Google-t, és egy általuk konfigurált ODP taxonómián működő Lemur-alapú

3. táblázat HITEC futások hatékonysága a 135 keresőkifejezésen

Futás	F_1 1. címkéző	F_1 2. címkéző	F_1 3. címkéző	Átlag pontosság	Átlag felidézés	Átlag F_1	Nincs címke
R1	12,47%	10,03%	10,82%	**13,55%**	9,58%	**11,11%**	**39**
R2	16,61%	14,48%	14,22%	**16,95%**	13,90%	**15,10%**	**31**
R3	17,43%	15,43%	16,39%	**19,36%**	14,53%	**16,42%**	**32**

4. táblázat HITEC futások hatékonysága azon keresőkifejezéseken, amelyre adott választ

Futás	Válaszok száma	F_1 1. címkéző	F_1 2. címkéző	F_1 3. címkéző	Átlag pontosság	Átlag felidézés	Átlag F_1
R1	**96**	15,19%	11,68%	13,06%	**13,55%**	13,35%	**13,31%**
R2	**104**	19,33%	16,13%	16,49%	**16,95%**	18,06%	**17,32%**
R3	**103**	20,50%	17,41%	19,18%	**19,36%**	19,10%	**19,03%**



6. ábra Az HKUST osztályozójának vázlata

keresőt[7], összesen mintegy 40 millió weboldalt és 50 GB-nyi adatot letöltve. Az ismertetett megoldásunkhoz analóg módon, az eredeti keresőkifejezéseket a keresőkhöz elküldve, a kapott eredmény oldalainak feldolgozásával készítettek a következő komponens, az osztályozók számára tanulóadatot. Nyilván itt is szükség van a három forrástaxonómia és a céltaxonómia kategóriái közötti leképezés meghatározására ahhoz, hogy az összegyűjtött tanulóadatokat alkalmazni lehessen a feladatra.

Ezt a leképezést két lépésben hozták létre. Elsőnek használtak egy, az általunk bemutatott kulcsszó illeszkedéses technikához hasonló (bár nem iteratív) algoritmust, ahol szintén a Wordnetet használták fel a célkategóriák nevének bővítésére. Ez a módszer nagy pontosságú leképezést biztosít, azonban a fedés (másképp felidézés) szempontjából kedvezőtlen, hiszen kevés olyan forráskategória van, amelyet közvetlenül a neve alapján egy célkategóriához lehet rendelni. Ezért a második lépésben SVM (szupport vektor gép) alapú tanulómódszert alkalmaztak, ahol a letöltött oldalakhoz az első lépésben rendelt célkategóriák jelentették a tanulóadatokat. Ez a korpusz összesen 15 millió weboldalt tartalmazott. Az így felépített vektortér modell már elegendően teljes lett, és így a fedés is kielégítőnek bizonyult.

A leglényegesebb különbség az általunk javasolt módszerhez képest a következő lépésben rejlik. Nyilván az előző két lépésben létrehozott különböző leképezések különböző osztályozó függvényeket és eltérő eredményeket adnak, amelyek bizonyos kategóriákon jobb teljesítménnyel működnek. Célszerű tehát ezeknek az *osztályozóknak* valamilyen *kombinációját* tekinteni oly módon, hogy az adott keresőkifejezésre a lehető legjobb eredményt kapjuk. Az osztályozók jóságának kiértékelését a 111 validációs adat segítségével valósították meg, amely alapján súlyfaktorokat rendeltek az osztályozókhoz. Amennyiben egy keresőkifejezést helytelenül osztályozott a rendszer, akkor a megfelelő súlyok állításával elérték, hogy a hibát kiküszöböljék. Ezzel a boosting jellegű [4] iteratív technikával az F_1 értéke jelentősen javítható. Mivel azonban a 111 adat nagyon kevés, ezért ez a módszer rendkívül hajlamos a túltanulásra. Ezt kiegyensúlyozandó olyan osztályozókat is bevettek a kombinációba, amelyek uniform súlyokkal rendelkeztek (statisztikai osztályozó). A végső eredményt a két típusú osztályozók eredményeinek kombinációjaként állították elő.

5.2. Osztályozás logikai regresszióval

Az F_1 alapú osztályozási hatékonyság versenyének második helyezését a floridai A.I. Insight, és MEDai cégek, valamint a berlini Humboldt Egyetem közös csapata érte el.

Módszerük [8] első lépéseként a Google keresőjét alkalmazták az ODP taxonómián[8] némileg módosított keresőkifejezéseken. Itt kihasználták a keresőnek helyesírási hibák javítására vonatkozó szolgáltatását (*„did you mean:...”*). Az ODP taxonómia és a céltaxonómia kategóriái közti leképezést manuálisan valósították meg, általában az ODP taxonómia felső két szintjének csomópontjait rendelték célkategóriákhoz, de ahol a finomítás megkívánta, akár a negyedik szintre is lementek a forrástaxonómiában. Az ennél is lejjebb lévő kategóriák besorolásához egy ajánló rendszert fejlesztettek ki, aminek eredményeképpen a manuális leképezést kiegészítették. A végső leképezés *n:m* típusú volt, ahol egy ODP kategóriához legfeljebb 3 célkategóriát rendeltek.

Az osztályozást az A.I. Insight logikai regresszión alapuló szoftverével végezték[9]. A modellben három paramétertípus van: a kategóriákra vonatkozó súlytényező, a kategória rangja az adott célkategóriák sorrendjében, illetve a két legjobb kategória közti súlytényezők

7 Lemur: http://www.lemurproject.org, ODP (Open Directory Project): http://dmoz.com
8 http://www.google.com/dirhp
9 Mitch (Multiple Intelligent Tasking Computer Heuristics)

különbsége. A szoftver ezeket a paramétereket egyetlen valószínűségi változóvá kombinálja, és ez alapján végzi a következtetést. A módszer előnye, hogy lehetőséget biztosít a pontosság és a felidézés közti optimalizásásra.

6. Összefoglalás

Jelen munkánkban bemutattuk a KDD kupa 2005-ös kiírására kifejlesztett algoritmusunkat. A megoldásnak két lényeges tényezője volt: egyrészt az Internet alapján megtalált és feldolgozott forrástaxonómiák, valamint a céltaxonómia közti leképezés meghatározása, és ily módon a tanulóadatok biztosítása; másrészt pedig a HITEC osztályozó hatékonysága. A Fürkész algoritmus más problémák megoldásában is használható, ahol különböző taxonómiák közt kell leképezést meghatározni, például különböző besorolási rendszert használó szabadalmi hivatalok alkalmazhatják a taxonómiák összehangolására.

Úgy érezzük, hogy eljárásunk sikeresen szerepelt, tekintve hogy először vettünk részt a KDD kupán, kezdetben a HITEC-en kívül nem állt rendelkezésre más segédeszköz, valamint hogy a díjazott csapatok közül a mienknek volt a legkevesebb tagja.

Köszönetnyilvánítás

Tikk Domonkost az MTA Bolyai János kutatói ösztöndíja támogatta. Jelen anyag elkészítését a Mobil Innovációs Központ is támogatta.

Irodalom

[1] HITEC categorizer online.
http://categorizer.tmit.bme.hu

[2] M. F. Porter:
An algorithm for suffix stripping.
Program, 14(3):130–137., July 1980.
http://telemat.det.unifi.it/book/2001/wchange/download/stem_porter.html

[3] G. Salton, M. J. McGill:
An Introduction to Modern Information Retrieval.
McGraw-Hill, 1983.

[4] R. E. Schapire, Y. Singer, A. Singhal:
„Boosting and Rocchio applied to text filtering”,
Proc. of SIGIR-98, 21st ACM Int. Conference on Research and Development in Information Retrieval, pp.215–223., Melbourne, Australia, 1998.
http://citeseer.ist.psu.edu/schapire98boosting.html

[5] Shen et al:
An ensemble search based method for query classification.
http://q2c.cs.ust.hk/q2c/Readme.pdf

[6] D. Tikk, Gy. Biró, J. D. Yang:
„A hierarchical text categorization approach and its application to FRT expansion”,
Australian Journal of Intelligent Information Processing Systems, 8(3):123–131., 2004.

[7] D. Tikk, Gy. Biró, J. D. Yang:
„Experiments with a hierarchical text categorization method on WIPO patent collections”,
In: N. O. Attok-Okine and B. M. Ayyub, editors, Applied Research in Uncertainty Modelling and Analysis, no.20, International Series in Intelligent Technologies, pp.283–302., Springer, 2005.

[8] D. Vogel et al,
„Classifying search engine queries using the web as background knowledge”,
SIGKDD Explorations (megjelenés alatt).
http://www.medai.com/publications/pdf/vogel_kddcup_2005.pdf