KIELIPANKKI JA LEMMIE-OHJELMISTO

Mickel Grönroos
CSC – Tieteellinen laskenta Oy, Espoo
Mickel.Gronroos@csc.fi

Manne Miettinen
CSC – Tieteellinen laskenta Oy, Espoo
Manne.Miettinen@csc.fi

CSC – Tieteellinen laskenta Oy:n ylläpitämän Kielipankin korpustyökalu Lemmie tarjoaa kielitieteilijöille ja kieliteknologeille mahdollisuuden tarkastella Kielipankin suomen ja suomenruotsin tekstikokoelmia nopeasti ja varsin monipuolisesti verkoselaimessa ja omissa ohjelmissaan. Tässä artikkelissa esitetään syyskuussa 2002 julkaistua Lemmie-ohjelmiston toista versiota.

Avainsanat: *korpuslingvistiikka, kieliresurssit, korpustyökalu, kielitieteellinen hakukone*

TAUSTAA

Opetusministeriön omistama tieteen tietotekniikan keskus CSC on palvellut kielitieteen ja kieliteknologian tutkijoita vuodesta 1997. Ensimmäinen kielitieteen tutkimusaineisto asetettiin tutkijoiden käyttöön vuonna 1998, kun Kotimaisen kielen tutkimuskeskuksen ja Helsingin yliopiston yleisen kielitieteen laitoksen yhdessä koostama Suomen kielen tekstipankki, versio A oli saatu valmiiksi. Tämä EU:n rahoittamassa LE PAROLE-hankkeessa koostettu aineisto oli laajuudeltaan 21,3 miljoonaa sanaa.

Vuosina 1999–2000 CSC osallistui yhdessä Kotimaisten kielten tutkimuskeskuksen ja Helsingin yliopiston yleisen kielitieteen laitoksen kanssa Suomen kielen tekstipankin laajennushankkeeseen, joka toteutettiin opetusministeriön Tieteen tietokantojen toimenpideohjelmasta saadun 450 000 mk (75 000 euroa) suuruisen määrärahan turvin. Laajennettuun aineistoon saatiin sisällytetyksi vuonna 1999 Joensuun yliopiston yleisen kielitieteen oppiaineen keräämä mittava sanomalehti Karjalaisen 34 miljoonan sanan laajuinen aineisto. Hankkeen tavoitteena oli laajentaa tutkijoiden käytössä oleva tutkimusaineisto 20 miljoonasta sanasta 100 miljoonaan sanaan tai ylikin. Tavoitteessa onnistuttiin erinomaisesti: vuonna 2000 oli kerätty ja koodattu yli 200 miljoonaa sanaa suomen ja suomenruotsin tekstejä. Noin puoleen teksteistä oli lisäksi liitetty automaattisesti morfosyntaktinen koodaus. Tekstimassan kasvu sekä koodauksen monimutkaistumisen aiheutti sen, etteivät perinteiset Unix-työkaluohjelmat kuten grep, sed ja awk enää soveltuneet suoraviivaisesti tekstien käsittelyyn.

Yhtenä laajennushankkeen tuloksena olikin Helsingin yliopiston yleisen kielitieteen

Kirjoittajien osoite:
CSC – Tieteellinen laskenta Oy
PL 405, 02101 Espoo

laitoksella suunniteltu ja toteutettu *Lemmie*-korpustyökalu (Grönroos 2000), joka tunnetaan nykyään nimellä *Lemmie-prototyyppi*. Graafisten Windows-sovellusten toimintalogiikkaan tottunut tutkija pystyi sen avulla helposti poimimaan sanamuototaajuuksia, KWIC-konkordansseja ja sanamuotojen kollokaatioita. Pintamuodon lisäksi tutkija kykeni hakemaan sanamuotoesiintymiä perusmuodon, sanaluokan ja sanamuodon taivutustietojen, esimerkiksi sijamuodon tai luvun avulla.

Lemmie-prototyypin käyttöönottoa haittasi kuitenkin käyttöliittymän riippuvuus Unixin ikkunointiteknologiasta (X Window System), joka edellyttää että käyttäjän työasemaan on asennettu erityinen palvelinohjelmisto. Kun Lemmie-ohjelmiston kehitystyö siirtyi vuonna 2001 CSC:hen Lemmien käyttöliittymä päätettiin toteuttaa Unixin ikkunointiteknologiasta riippumattomalla tavalla. Kesällä 2001 saatiin valmiiksi *Lemmieshell*-niminen komentotulkkikäyttöliittymä, joka toimii tavallisessa terminaali-ikkunassa. Ohjelman perustana oli CSC:ssä suunniteltu ja toteutettu Lemmie-ohjelmointirajapinta ja leksikaalinen tietokanta. Myöhemmin samana vuonna kehitettiin saman ohjelmointirajapinnan ja tietokannan päälle selainpohjainen käyttöliittymä, *WWW-Lemmie,* joka liitettiin osaksi CSC:n Tutkijan käyttöliittymä -nimistä verkkopalvelua.

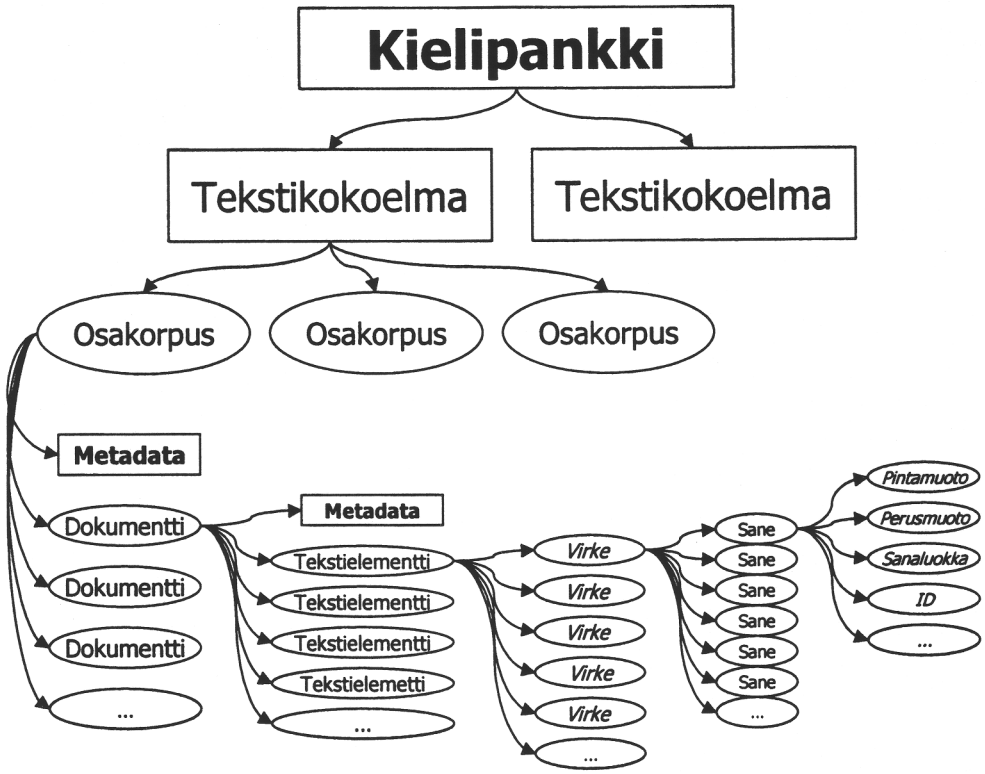
WWW-Lemmien ensimmäinen versio oli lokitietojen valossa odotettu uudistus Unixia vierastaville tutkijoille; vilkkaimpana viikkona tehtiin yli 500 hakua. Toisaalta esitettiin myös paljon parannusehdotuksia ja toiveita uusista toiminnoista. Niinpä CSC:ssä palattiin talvella 2002 suunnittelupöydän ääreen ja keväällä 2002 ryhdyttiin toteuttamaan uutta versiota, johon kuului parannettu ohjelmointirajapinta sekä monipuolisempi www-käyttöliittymä. Lemmie 2 julkistettiin syyskuussa 2002.

LEMMIE 2:N RAKENNE

Lemmie 2 koostuu leksikaalisesta tietokannasta, jota käsitellään Perl-ohjelmointikielellä toteutetun olioperustaisen ohjelmointirajapinnan (Lemmie API 2) kautta. Verkkoselaimella käytettävä käyttöliittymä WWW-Lemmie 2.0 on rakennettu ohjelmointirajapinnan päälle.

Lemmie API 2 ohjelmointirajapinnassa korpuslingvistinen hakukone ja siihen liittyvät käsitteet on mallinnettu olioperustaisen ajattelun mukaisesti ns. olioina ja niiden välisinä suhteina. Esimerkiksi Kielipankki on mallinnettu hierarkisena järjestelmänä, joka koostuu tekstikokoelmista, jotka koostuvat osakokoelmista, jotka koostuvat metadatasta (esimerkiksi sanemäärä) ja dokumenteista. Dokumentit koostuvat metadatasta (esimerkiksi julkaisupäivämäärä) ja juoksevasta tekstistä, joka koostuu tekstielementeistä (esimerkiksi otsikot, kuvatekstit, ingressit ja kappaleet), jotka koostuvat virkkeistä, jotka ovat järjestetty jono saneita (sanamuotoesiintymiä), jotka koostuvat sanamuotopiirteistä, kuten esimerkiksi pintamuodosta, perusmuodosta ja sijamuodosta. (Kaavio 1).

KAAVIO 1. Kielipankin aineistohierarkia.



Lemmie API 2 mallintaa edellä esitetyn hierakian olioperustaisessa ohjelmoinnissa käytettyjen (olio)luokkien avulla. Lemmie-luokka edustaa laajinta kokonaisuutta, "korpuslingvististen hakukoneiden" luokkaa, josta käsin muita luokkia käytetään. Leksikaalinen tietokanta ja sen tarjoamat palvelut on myös abstrahoitu omaksi luokakseen (DatabaseConnection), jotta tulevaisuudessa olisi mahdollisimman helppo vaihtaa tietokannan teknistä toteutusta. Käyttäjän omia koodauksia varten on olemassa oma luokkansa (Luaf), johon voi tallettaa koodausvirheiden korjauksia tai omaa koodausta (esimerkiksi semanttista luokittelua).

Lemmie-luokan avulla voidaan luoda yksi tai useampi korpusluokan olio (Corpus). Korpusluokka vastaa joko yhtä tekstikokoelmassa esimääriteltyä osakorpusta (esimerkiksi Turun Sanomat 1999) tai yhtä käyttäjän itse määrittelemää osakorpusta. Korpusluokan oliot koostuvat metadatasta ja dokumenteista. Dokumentit kuvataan dokumenttiluokan (Document) olioina. Niihin kuuluu metadataa (esimerkiksi julkaisupäivämäärä ja tekijä) ja saneita. Saneet on kuvattu sanaluokan (Token) olioina, joilla on sanamuotopiirteiden lisäksi yksiselitteinen tunniste.

Dokumentin ja saneen välisiä tasoja (virkkeitä, tekstikappaleita, lukuja jne.) ei ole kuvattu omina luokkinaan, mutta niitä voi rajoitetusti käsitellä konkordanssi- ja kollokaattiluokista käsin (esimerkiksi konkordanssin tai kollokaation kontekstiksi voi määritellä virkkeen tai kappaleen).

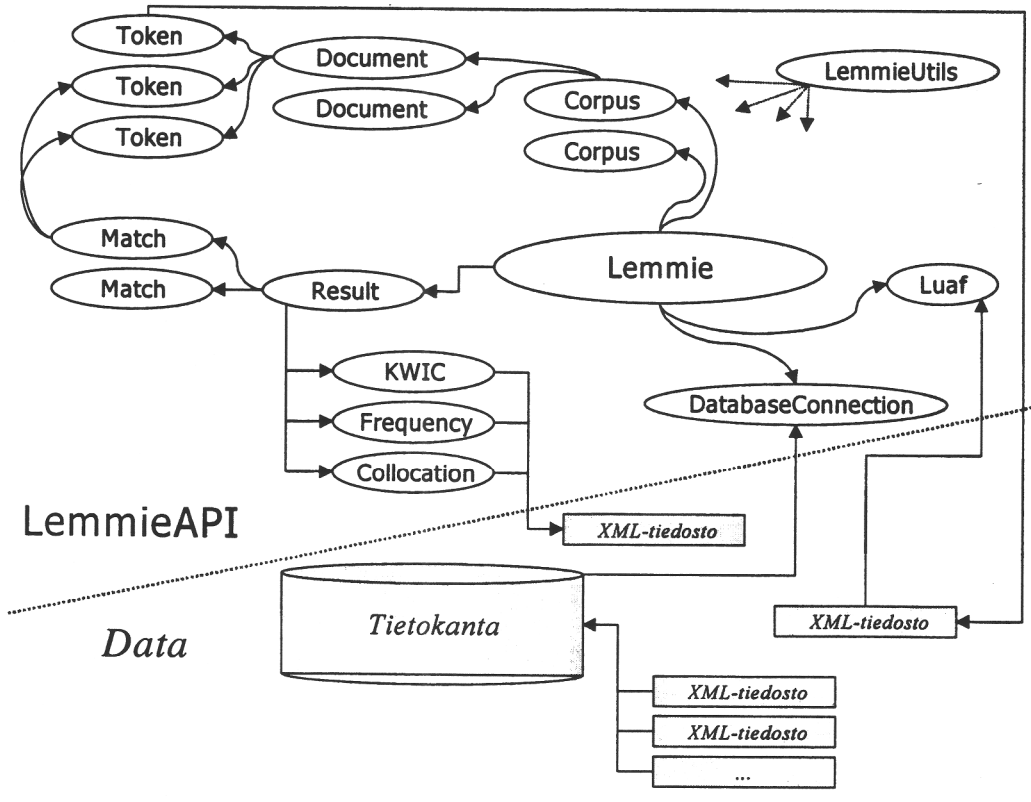
Hakutulosten käsittelyä varten on määritelty omia olioluokkia. Tulosluokka (Result) on näistä perustavanlaatuisin. Kun Lemmie-luokan olion query-metodilla on tehty haku

joukkoon osakorpuksia, tietokantaluokka palauttaa tulosluokan olion.

Tulosluokan olio sisältää noodiluokan (Match) olioita. Jokainen noodiolio vastaa yhtä hakulausekeeseen täsmäävää joukkoa saneolioita. Tulosoliota ei voi tarkastella suoraan, vaan tulostamista varten tulosolio on ensin muutettava haluttuun muotoon: KWIC-konkordanssiksi (Concordance-luokan olioksi), frekvenssitaulukoksi (Frequency-luokan olioksi) tai kollokaatiotaulukoksi (Collocation-luokan olioksi). Näimä oliot voidaan esittää merkkijonona tai XML-dokumenttina.

Lemmie API 2:n oliorakenne kuvataan kaaviossa 2. Kaarinuolimerkintä luetaan *A koostuu B:stä*, kulmanuoli *A:sta saadaan B*.

KAAVIO 2. Lemmie API 2:n oliorakenne.



LEMMIEN TOIMINNALLISUUS

Lemmien syvin olemus on suorittaa *korpuslingvistisen hakukoneen* tehtäviä. Lemmiellä voi hakea Kielipankin suomen ja suomenruotsin tekstikokoelmista saneita ja saneiden yhdistelmiä ja tarkastella niitä kontekstissa KWIC-konkordanssina, tulosten taajuuksia frekvenssitaulukossa tai tuloksessa olevien sanamuotojen kollokaatioita kollokaatiotaulukossa (kts. esimerkit 1, 2, 3 ja 4 alla).

On tärkeä muistaa, että sanan pintamuoto on vain yksi sanamuodon piirteistä, vaikka useimmat toimenpiteet kohdistuvat oletusarvoisesti pintamuotoon. Käyttäjä voi silti yhtä lailla hakea millä tahansa sanamuodon

piirteellä, kuten esimerkiksi perusmuodolla, sanaluokalla, sijamuodolla tai muulla sanamuodon taivutuspiirteellä. Samaan tapaan myös tuloksia voi tarkastella minkä tahansa piirteen mukaan. Frekvenssitaulukko voidaan esimerkiksi yhtä hyvin laskea tuloksen sisältämien sijamuotojen, kuin pintamuotojen perusteella.

Hakutoiminnon lisäksi Lemmiellä voi myös verrata kahta tulosta keskenään. Saadessaan kaksi hakutulosta, testi- ja viitetuloksen, Lemmie hakee testituloksesta sellaiset sanamuotojen piirteet, jotka ovat tilastollisesti merkittäviä testituloksessa. Tällaisia ovat esimerkiksi sellaiset sanamuotojen perusmuodot, jotka esiintyvät ainoastaan testituloksessa, tai jotka esiintyvät yli kaksi kertaa useammin testituloksessa kuin viitetuloksessa perusmuotojen suhteellisia frekvenssejä vertaillessa. Tätä toimintoa hyödyntäen voi esimerkiksi tarkastella minkälaisia verbejä esiintyy *mies*-perusmuodon yhteydessä, verrattuna *nainen*-perusmuodon verbeihin jossain korpuksessa (kts. esimerkki 5 alla). Lemmiellä voi myös vertailla testituloksen ja viitetuloksen sanamuotojen piirteiden kollokaatioita.

Lemmien avulla on myös mahdollista koostaa oma korpus olemassa olevien korpusten dokumenteista. Tätä toimintoa hyödyntämällä käyttäjät voivat tarkasti määritellä, mihin dokumentteihin he kohdistavat hakunsa.

Lisäksi Lemmiellä voi liittää korpuksissa oleviin saneisiin omaa mielivaltaista sanekohtaista koodausta. Tämä on käytännöllistä, jos käyttäjä haluaa varmistaa, että käytössä oleva korpus on koodattu juuri niin kuin tutkimusongelma edellyttää. Dokumenteissa oletuksena oleva kieliopillinen koodaus on tehty automaattisesti Kielikone Oy:n toimittamalla Textmorfo- ja Lingsoft Oy:n toimittamalla SWECG-ohjelmistolla, eikä tuloksia ole korjattu käsin. Näin ollen oletuskoodauksessa on väistämättä virheellisiä luentoja. Liittämällä saneisiin korjatut koodaukset käyttäjä voi välttää automaattisen koodauksen virheet.

Kieliopillisen koodauksen korjaamisen lisäksi samaa tekniikkaa voi myös hyödyntää omien piirteiden, esimerkiksi semanttista luokittelua, lisäämiseksi saneihin. Saneolioille on määritelty valmiiksi *extra*-niminen attribuutti, joihin käyttäjä voi liittää mielivaltaista sanekohtaista tietoa. Tosin saneille on myös mahdollista assosioida muunkin nimisiä piirteitä oman harkinnan mukaan.

Käyttäjän omat korjaukset ja lisäykset saneiden piirteisiin tallentuvat käyttäjän määrittelemään XML-tiedostoon, jonka sisältämät saneet täydentävät tai korvaavat leksikaalisessa tietokannassa olevien samoilla tunnisteilla varustettujen saneiden piirteet. Näin ollen korjaukset ja lisäykset ovat haettavissa tavalliseen tapaan, vaikka ne haetaankin käyttäjän määrittelemästä XML-tiedostosta, eikä leksikaalisesta tietokannasta. Haun voi tehdä WWW-Lemmie 2.0:lla tai omassa ohjelmassa ohjelmointirajapinnan kautta. Itse piirteiden korjaus ja lisäys ei onnistu vielä suoraan WWW-Lemmiestä käsin, mutta Kielipankissa on koekäytössä komentotulkkityyppinen ict-niminen työkalu koodauksen vuorovaikutteiseen muokkaukseen.

ESIMERKKEJÄ WWW-LEMMIE 2.0:N KÄYTÖSTÄ

ESIMERKKI 1. WWW-Lemmie 2.0:lla luotu KWIC-konkordanssi, jossa esiintyy sellaiset Iltalehti 1996-osakorpuksessa olevat verbit, joita seuraa sane, jolla on pintamuoto *pakoon*. Tuloksessa näytetään saneiden **pintamuoto**. (Hakulauseke: [pos='Verb'][wf='pakoon'])

1	mutta tosiasiassa hän vain	**juoksi pakoon**	henkensä edestä . ----	ilta1996
2	vankeustuomiotaan kärsivä Timo Punkkinen	**lähti pakoon**	muurin viereisen itäisen selliosaston	ilta1996
3	hälytti poliisit . Ryöstäjä	**lähti pakoon**	yhdessä häntä pihalla odottaneen	ilta1996
4	Kun pääset irti ,	**pinkaise pakoon**	. Jätä korkokengät sinne	ilta1996
5	hyökkääjä on sekava ,	**pyri pakoon**	ja huuda vasta sitten apua	ilta1996
6	päästä hän kääntyy ja	**pyrähtää pakoon**	. Hän ei uskalla	ilta1996
7	kaatui , jolloin tyttö	**pääsi pakoon**	. Kajaanin poliisi etsii	ilta1996
8	, kun on pitänyt	**tulla pakoon**	vessaan ? Kysymykseen löytyy	ilta1996

ESIMERKKI 2. WWW-Lemmie 2.0:lla luotu frekvenssitaulukko, jossa näytetään sellaiset Turun Sanomat-lehden osakorpuksissa olevat verbit, joita seuraa sane, jolla on pintamuoto *pakoon*. Tulokset on laskettu saneiden **perusmuotojen** mukaan, jotka myös näytetään taulukossa. Eri osakorpuksissa olevat saman perusmuotopiirreyhdistelmän frekvenssitiedot on yhdistetty taulukossa yhdeksi luvuksi. Ensimmäisessä sarakkeessa on rivin perusmuotopiirreyhdistelmän absoluuttinen frekvenssi, toisessa yhdistelmän suhteellinen frekvenssi ja kolmannessa perusmuotojen yhdistelmä. Taulukon ensimmäisellä rivillä on kaikkien hakulauseketta vastaavien perusmuotojen yhdistelmien frekvenssit yhteenlaskettuna. Sellaiset yhdistelmät, jotka esiintyvät harvemmin kuin kaksi kertaa on poistettu. (Hakulauseke: [pos='Verb'][wf='pakoon'])

Joint Frequency (tusa1998 + tusa1999)

Abs	Rel	Token
121	0,0000052105	[pos='Verb'][wf='pakoon']
33	0,0000014211	[bf='lähteä'][bf='pako']
32	0,0000013780	[bf='päästä'][bf='pako']
17	0,0000007321	[bf='juosta'][bf='pako']
9	0,0000003876	[bf='pötkiä'][bf='pako']
3	0,0000001292	[bf='ehtiä'][bf='pako']
2	0,0000000861	[bf='ajaa'][bf='pako']
2	0,0000000861	[bf='pyrkiä'][bf='pako']
2	0,0000000861	[bf='voida'][bf='pako']

Note: 21 rows not showing (did not beat minimum scores).

ESIMERKKI 3. WWW-Lemmie 2.0:lla luotu frekvenssitaulukko, jossa näytetään sellaisten Karjalainen-lehden osakorpuksissa 1992, 1995 ja 1998 olevien sanamuotojen perusmuodot, joiden perumuotopiirre alkaa merkkijonolla *luomu*, mutta joiden perumuoto ei ole *luomus*. Tulokset on laskettu näiden **perusmuotojen** mukaan, jotka myös näytetään taulukossa. Frekvenssit näytetään osakorpuksittain ja ne tulokset, jotka esiintyvät harvemmin kuin kymmenen kertaa (siinä osakorpuksessa missä esiintymiä on eniten) on poistettu. Oletuksena on, että voimme tässä taulukossa nähdä kuinka suhteellisen tavallisten *luomu*-sanojen käyttö on lisääntynyt 1990-luvun aikana. (Hakulauseke: [bf='luomu*' bf!='luomus'])

Joint Frequency (karj1992 + karj1995 + karj1998)			karj1992		karj1995		karj1998	
Abs	**Rel**	**Token**	**Abs**	**Rel**	**Abs**	**Rel**	**Abs**	**Rel**
1092	**0,0000659238**	**[bf="luomu*" bf!="luomus"]**	**72**	**0,0000043466**	**372**	**0,0000224576**	**515**	**0,0000310904**
136	0,0000082103	[bf='luomutuotanto']	8	0,0000018117	57	0,0000092250	71	0,0000118926
121	0,0000073047	[bf='luomutuote']	9	0,0000020382	48	0,0000077684	64	0,0000107201
80	0,0000048296	[bf='luomuviljely']	11	0,0000024911	43	0,0000069592	26	0,0000043550
52	0,0000031392	[bf='luomutila']	5	0,0000011323	28	0,0000045316	19	0,0000031825
44	0,0000026563	[bf='luomumaito']	1	0,0000002265	28	0,0000045316	15	0,0000025125
41	0,0000024752	[bf='luomu']	4	0,0000009059	10	0,0000016184	27	0,0000045225
40	0,0000024148	[bf='luomuvilja']	1	0,0000002265	14	0,0000022658	25	0,0000041875
37	0,0000022337	[bf='luomuviljelijä']	1	0,0000002265	17	0,0000027513	19	0,0000031825
19	0,0000011470	[bf='luomuruis']	5	0,0000011323	10	0,0000016184	4	0,0000006700
19	0,0000011470	[bf='luomuruoka']	0	0	1	0,0000001618	18	0,0000030150
18	0,0000010867	[bf='luomuleipä']	0	0	4	0,0000006474	14	0,0000023450
17	0,0000010263	[bf='luomua']	1	0,0000002265	3	0,0000004855	13	0,0000021775
14	0,0000008452	[bf='luomuala']	0	0	10	0,0000016184	4	0,0000006700
14	0,0000008452	[bf='luomujauho']	1	0,0000002265	8	0,0000012947	5	0,0000008375
14	0,0000008452	[bf='luomumeijeri']	0	0	14	0,0000022658	0	0
14	0,0000008452	[bf='luomun']	0	0	4	0,0000006474	10	0,0000016750
14	0,0000008452	[bf='luomuun']	0	0	3	0,0000004855	11	0,0000018425
12	0,0000007244	[bf='luomuolut']	0	0	0	0	12	0,0000020100
11	0,0000006641	[bf='luomu-liitto']	0	0	3	0,0000004855	8	0,0000013400
10	0,0000006037	[bf='luomu-tuotanto']	5	0,0000011323	4	0,0000006474	1	0,0000001675
10	0,0000006037	[bf='luomupuuro']	0	0	0	0	10	0,0000016750
10	0,0000006037	[bf='luomuviljelmä']	0	0	1	0,0000001618	9	0,0000015075

ESIMERKKI 4. WWW-Lemmie 2.0:lla luotu kollokaatiotaulukko, joissa näytetään Karjalainen-lehden osakorpuksissa olevien *pakoon*-pintamuodon kollokaatiot. Laskussa on käytetty mutual infomation- ja t-score –kaavaa. Kollokaattien osalta lasketaan **perusmuodon** esiintymiä ja noodin osalta **pintamuodon** esiintymiä. Lisäksi vaaditaan, että laskettavien kollokaatioiden pitää esiintyä vähintään viisi kertaa korpuksessa ja että kaikki kollokaation osat esiintyvät saman virkkeen sisällä. Mutual information-kaavan mukaan saatujen arvojen pitää ylittää alimmaisraja 5. Kollokaatiotaulukko on järjestetty mutual information-kaavan arvojen mukaan niin, että vahvimman arvon saaneet kollokaatiot näytetään ensin. Taulukosta näemme, että *sodan* ja *pakkasen* vuoksi pakoon *pötkitään*, *sännätään*, *juostaan*, *lähdetään* ja joskus myös *päästetään*. (Hakulauseke: [wf='pakoon'])

Collocation Table

Collocation	mutual_information	t_score	Abs (Left/Right)	Rel	Node Abs	Collocate Abs
[bf='pötkiä'] pakoon	11,7521216986	2,9991306553	9 (7 / 2)	0,0000002512	377	62
[bf='sännätä'] pakoon	10,6855364469	2,6441450645	7 (6 / 1)	0,0000001953	377	101
[bf='juosta'] pakoon	6,7677951254	3,9632937359	16 (11 / 5)	0,0000004465	377	3489
[bf='oikeastaan'] pakoon	6,3226812095	2,7930905488	8 (8 / 0)	0,0000002233	377	2375
[bf='lähteä'] **pakoon**	6,1054653382	9,2446002055	88 (57 / 31)	0,0000024558	377	30370
[bf='sota'] pakoon	5,7924001465	4,3914454914	20 (19 / 1)	0,0000005581	377	8575
[bf='pakkanen'] pakoon	5,4767781513	2,1858557856	5 (4 / 1)	0,0000001395	377	2668
[bf='päästä'] pakoon	5,3816144466	7,8081124347	64 (57 / 7)	0,0000017860	377	36479

ESIMERKKI 5. Alla olevissa taulukoissa vertaillaan *mies*-perusmuodon oikealla puolella esiintyviä verbejä *nainen*-perusmuodon vastaaviin verbeihin. (Tulokset on luotu hakulausekkeilla: [bf='mies'][pos='Verb'] ja [bf='nainen'][pos='Verb']) Ensimmäinen taulukko sisältää sellaiset testituloksen noodeissa olevat perusmuodot, jotka esiintyvät ainoastaan testituloksessa eikä siis ollenkaan viitetuloksessa. Toinen taulukko sisältää sellaiset testituloksessa olevat perusmuodot, jotka esiintyvät vähintään viisi kertaa enemmän testituloksessa, kuin viitetuloksessa. Lisäksi vaaditaan, että testituloksessa olevien perusmuotojen frekvenssi viitetuloksen noodeissa on vähintään viisi. Molemmat tulokset on generoitu Iltalehti 1996-korpuksesta.

UNIQUE - Single tokens (in nodes) only found in frequency table of bf_mies_pos_Verb_ilta1996.lres

Abs	Rel	Token
887	0,0009281699	[bf='mies']
15	0,0000156962	[bf='ampua']
10	0,0000104641	[bf='ottaa']
9	0,0000094177	[bf='menehtyä']
8	0,0000083713	[bf='ehtiä']
7	0,0000073249	[bf='ryöstää']
6	0,0000062785	[bf='ajaa']
5	0,0000052321	[bf='alkaa']
5	0,0000052321	[bf='lyödä']

OVERREPRESENTED single tokens (in nodes) in frequency table of bf_mies_pos_Verb_ilta1996.lres

Quotient	Token
7,500	[bf='kuolla']
7,000	[bf='sanoa']
7,000	[bf='viedä']
6,000	[bf='jäädä']
6,000	[bf='päästä']
5,000	[bf='yrittää']
5,000	[bf='vangita']
5,000	[bf='tuntea']
5,000	[bf='epäillä']
5,000	[bf='loukkaantua']
5,000	[bf='onnistua']
4,000	[bf='kertoa']
3,500	[bf='lähteä']
3,375	[bf='saada']

ESIMERKKI LEMMIE API 2:N KÄYTÖSTÄ

Lemmien ohjelmointirajapinnan avulla Kielipankin käyttäjä voi kirjoittaa omia Perl-ohjelmia, jotka käyttävät leksikaalista tietokantaa ja hyödyntävät WWW-Lemmie 2.0:sta tuttuja toimintoja. Alla oleva ohjelma on yksinkertainen tekstipohjainen konkordanssiohjelma, joka saa syötteen komentoriviltä ja tulostaa konkordanssirivit standarditulosteeseen (STDOUT). Konkordanssirivit muunnetaan 80-merkin levyisiksi, jotta ne täyttäisivät normaalikokoisen terminaali-ikkunan koko leveydeltään.

```
#!/usr/bin/perl

## kwic.pl
##
## © 2002 CSC - Tieteellinen laskenta Oy

use strict; # Käytetään "kurinalaista" syntaksia
use Lemmie; # Otetaan käyttöön Lemmie API

## Tarkistetaan, että ohjelma saa riittävästi parametrejä
## komentoriviltä
if ( scalar @ARGV < 2 )
{
    print STDERR "kwic hakulauseke korpus-ID [korpus-ID[, ...]]\n";
    exit;
}

## Otetaan talteen ensimmäinen parametri ja käytetään sitä
## hakulausekkeena
my $hakulauseke = shift;

## Luodan Lemmie-luokan olio ja tehdään muutama tarkistus
my $lemmieO = new Lemmie;
die "Lemmie-olion luonti epäonnistui!"
    unless ( ref( $lemmieO ) eq "Lemmie" );

## Oletetaan muiden komentoriviparametrien (@ARGV) olevan korpuksia.
## Otetaan ne käyttöön jos mahdollista
foreach my $corpusO ( $lemmieO->showAvailableCorpora )
{
    if ( &member( $corpusO->getId, \@ARGV ) )
    {
       $lemmieO->useCorpus( $corpusO ) ||
           die "Virhe! ", $lemmieO->errStr;
    }
}

## Tulostetaan STDOUT:iin käytössä olevat korpukset käyttäjälle
## tiedoksi
print "Käytössä olevat korpukset: ";
print ( join ( ", ", map {$_->getId} $lemmieO->showCorporaInUse ) );
print "\n";

## Tulostetaan käytettävä hakulauseke STDOUT:iin käyttäjälle tiedoksi
print "Hakulauseke: ", $hakulauseke, "\n";
```

```
## Tehdään haku tai tulostetaan virheilmoitus, mikäli haku ei onnistu
my $resultO = $lemmieO->query( $hakulauseke ) ;
die ( "Haku epäonnistui: ", $lemmieO->errStr )
    unless ( ref( $resultO ) eq "Lemmie::Result" );

## Tulostetaan tulosten määrä STDOUT:iin
print "Tuloksia: ", $resultO->hits, "\n";

## Suoritus loppuu tähän, jos tuloksia ei ole
exit if ( $resultO->hits == 0 );

## Luodaan konkordanssiolio. Tulosolio käytetään datana.
## Tulostetaan virheilmoitus, mikäli luominen ei onnistu
my $kwicO = $resultO->asKwic;
die ( "Konkordanssin luonti epäonnistui: ", $resultO->errStr )
    unless ( ref( $kwicO ) eq "Lemmie::Kwic" );

## Asetetaan kontekstin kooksi 9 sanaa noodin molemmin puolin, jotta
## voimme myöhemmin poistaa merkkejä molemmin puolin saadaksemme
## 80 merkkiä leveitä konkordanssirivejä. Muut konkordanssiolion
## muuttujat saavat jäädä oletusarvoihin
$kwicO->setLeftContext( 9 )  || die $kwicO->errStr;
$kwicO->setRightContext( 9 ) || die $kwicO->errStr;

## Haetaan konkordanssiin tuloksen kontekstit, jos mahdollista
$kwicO->calculate || die "Virhe! ", $kwicO->errStr;

## Extrahoidaan konkordanssi merkkijonona oliosta
## ja lisätään loppuun rivinvaihdon
my $merkkijono = $kwicO->asString."\n";

## Muutetaan rivejä niin, että ne ovat 80 merkkiä pitkiä
## (vasen konteksti = 35 merkkiä,
##  noodi + oikea konteksti =  45 merkkiä)
$merkkijono =~ s/[^\n]*?(.{35})\s{2}(.{45}).*?(\n)/$1 $2$3$4/g;
$merkkijono =~ tr/ //s;

## Tulostetaan muunnettu merkkijono STDOUT:iin
print $merkkijono;

## Valmis, ohjelman suoritus päättyy
exit;

## Tarvittava aliohjelma, jolla tarkistetaan, mikäli
## arvo esiintyy listassa
sub member
{
       my $cand = shift;
       my $rl   = shift;
       foreach (@$rl)
       {
                  return 1 if ($cand eq $_);
       }
       return 0;
}
```

TULEVAISUUS

WWW-Lemmie 2.0 ja Lemmie API 2 ovat toiminnallisuudeltaan merkittäviä parannuksia ensimmäiseen Lemmie-implementaatioon nähden, mutta parannettavaakin on. Kielipankkiin tule aika ajoin uusia ehdotuksia miten Lemmietä voisi parantaa tulevaisuudessa. Ja teknologian, erityisesti kieliteknologian, kehittyessä uusia haasteita tulee vastaan.

WWW-Lemmieen voisi lisätä toiminnon, jolla käyttäjä voisi automaattisesti koodata, rakenteellisesti ja kieliopillisesti, omia dokumenttejaan ja tulokset tallettaa itse määrittelemäänsä korpukseen Lemmien kautta käytettäväksi.

Uusien kieliteknologisten analyysiohjelmien (esimerkiksi syntaktisten jäsentimien tms.) kehittyessä voisi myös olla käytännöllistä lisätä olemassa oleviin korpuksiin analyysiohjelmien tuottamaa tietoa Lemmien kautta haettavaksi. Tällainen koodauksen lisääminen onnistuu jo osittain Lemmie API 2:n avulla (XML-muotoisten lisäystiedostojen avulla), mutta hakeminen muuttuu hitaaksi, kun lisäystiedostojen koko kasvaa suureksi.

Lemmien hakunopeutta kannattaisi myös optimoida. Voisi tutkia uusien tietokantatekniikoiden tarjoamia mahdollisuuksia sekä etsiä keinoja olemassa olevan tietokannan rakenteen tehostamiseksi. Uusien tietokantojen käyttö vanhojen kanssa rinnakain ei pitäisi aiheuttaa käyttäjille näkyviä muutoksia, sillä tietokannan kanssa käyty keskustelu on ohjelmointirajapinnassa eriytetty tietokantayhteys-luokkaan. Uuden tietokannan saisi siis käyttöön kirjoittamalla rajapinnan vaatimat metodit toteuttavan tietokantayhteysluokan.

THE LANGUAGE BANK OF FINLAND AND THE LEMMIE SOFTWARE

Mickel Grönroos, CSC – Scientific Computing Ltd., Espoo

Manne Miettinen, CSC – Scientific Computing Ltd., Espoo

The corpus query tool Lemmie offers researchers in the fields of linguistics and language technology an opportunity to make quick and versatile searches in the Finnish and Finland-Swedish text collections of the Language Bank of Finland maintained at CSC – Scientific Computing Ltd. The second version of Lemmie released in September 2002 is presented in this article.

Keywords: corpus linguistics, language resources, corpus query tool, linguistic