

# Digitaaliset ihmistieteet kielentutkimuksessa: tekstinlouhinta

Työpaja

# Päivän aikataulu ja ohjelma

- Esittely (nämä kalvot). Mitä on tekstinlouhinta ja miten se liittyy kielentutkimukseen?
- Hands-on 1: Suomi24:n aihepiirit ja niiden louhinta Topic modelingin avulla
- Ruokatauko
- Hands-on 2: Miten köyhyydestä puhutaan, ja mitä diskursseja siihen liitetään? Ohjattua koneoppimista ja avainsana-analyysiä.
- Ryhmätyöaiheen sopiminen ja valmistelu

# Tekstinlouhinta ja kielentutkimus

- Kielentutkimuksessa konelukuisten aineistojen käytöllä pitkät perinteet
- *Brown Corpus of Contemporary American English* 1964
- Miljoona sanaa vuonna 1961 kirjoitettua amerikanenglantia, jaoteltuina 500 tekstikategoriaan
- Myöhemmin samoilla spekseillä
  - Lancaster-Oslo-Bergen (LOB) Corpus: brittienglantia vuodelta 1961
  - Frown 1990-luvun alun amerikanenglantia
  - FLOB 1990-luvun alun brittienglantia
- ... ja vielä myöhemmin vielä enemmän aineistoja

# Menetelmiä ja tutkimuskohteita korpuslingvistiikassa

## Tutkimusesimerkkejä

- englannin 1. ja 2. persoonan pronomien vaihtelu (Vartiainen et al. 2013)
  - • Naiset käyttävät enemmän kuin miehet
  - → ns. sukupuolettunut kirjoitustyyli; miesten tyyli usein informatiivisempi, naiset keskittyvät vuorovaikutukseen
- englannin, -s ja -th (esim. *has* vs. *hath*)

# Menetelmiä ja tutkimuskohteita korpuslingvistiikassa

- Kollokaatit eli sanojen yhteisesiintymät
  - “You shall know a word by the company it keeps” (Firth 1957)
  - ... kertovat sanan merkityksestä ja käytöstä
- *Naukua*-verbin subjekti usein eläin
- *Cause* kollokaateja mm, *abandonment*, *accident*, *alarm* ja *anger*  
→ tyypillisesti negatiivinen
- vrt. Jarmo Jantusen esimerkit korpusavusteisesta diskurssianalyysistä!
- vrt. *naukua* Korpissa:  
[https://korp.csc.fi/#?stats\\_reduce=word&cqp=%5B%5D&corpus=ftb3\\_europarl,ftb3\\_jrcacquis,ftb2,s24\\_001,s24\\_002,s24\\_003,s24\\_004,s24\\_005,s24\\_006,s24\\_007,s24\\_008,s24\\_009,s24\\_010,s24,ylilauta,reittidemo&search=lemgram%7Cnaukua.vb.1&search\\_tab=2&word\\_pic](https://korp.csc.fi/#?stats_reduce=word&cqp=%5B%5D&corpus=ftb3_europarl,ftb3_jrcacquis,ftb2,s24_001,s24_002,s24_003,s24_004,s24_005,s24_006,s24_007,s24_008,s24_009,s24_010,s24,ylilauta,reittidemo&search=lemgram%7Cnaukua.vb.1&search_tab=2&word_pic)

# Menetelmiä ja tutkimuskohteita korpuslingvistiikassa

## Avainsana-analyysi

- Aineistossa ylliedustetut sanat verrattuna verrokkiaineistoon  
→ avainsanat
  - Kokonaiset tekstiaineistot → aineiston sisältö ja tyyli
  - (esim. Uutisissa aihepiirejä, raportointiin liittyviä verbejä)
  - Ns. Pätkestä kootut aineistot → diskurssit
  - (esim. Homodiskurssit)

**Esimerkki avainsanoista: Suomi24,  
homo-aineisto vs. hetero-aineisto**

F	AS-arvo	sanamuoto	F	AS-arvo	sanamuoto
1255	829.404	vitun	351	179.312	suomessa
1186	729.114	vittu	138	172.856	hesa
1416	524.196	kirkko	395	168.239	hommaa
1463	404.429	suomen	184	167.229	haista
1410	390.456	kirkon	1746	161.882	heitä
2385	368.306	jumalan	197	161.548	muslimit
708	324.149	paskaa	2203	158.086	vastaan
451	313.362	saatana	739	154.698	vihaa
524	291.411	saatanan	1014	150.307	jeesus
327	261.841	ateistit	651	145.305	kirkossa
1228	228.951	raamatun	364	140.416	tappaa
472	226.555	paska	427	136.456	kirkosta
2365	194.445	jumala			

# Kieliteknologiasta tekstinlouhintaan

- Edellä esitellyt menetelmät käytössä myös valmiissa korpustyökaluissa, kuten *Antconc* ja *Wordsmith*
- Menetelmillä on kuitenkin rajoituksensa
  - Vaativat käsin tehtävää ryhmittelyä (esim. *kirkko*, *jumala*, *raamatun* viittaavat samaan asiaan)
  - Käsittelevät aineistoja kokonaisuuksina, jolloin tekstienvälinen vaihtelu jää pimentoon
  - Eivät sovellu erittäin suuriin aineistoihin
  - Muutenkin hiukan kömpelöitä
  - → tekstinlouhinta seuraava askel!



# Mitä on tekstinlouhinta?

- Menetelmiä, joilla etsitään tietoa suuresta määrästä dataa
- Käyttää hyväkseen kieliteknologiaa ja koneoppimista
- Etu: joustavaa ja tehokasta
- Haitta: edellyttää ohjelmointitaitoja
- Näitä kaikkia opiskellaan kieliteknologian sivuainekokonaisuudessa
- Tässä työpajassa lyhyt esimerkki

# Koneoppimisen kaksi päämenetelmää

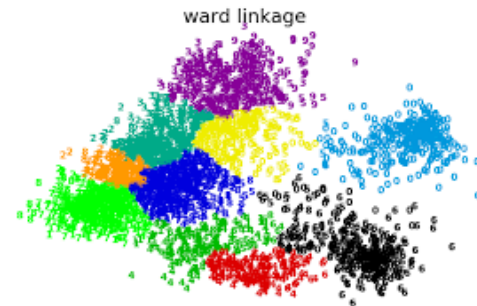
## *Ohjattu koneoppiminen*

- Perustuu (käsin tehtyyn) harjoitusaineistoon, jossa halutut luokat on merkitty
- Ohjelma opettelee luokittelemaan uusia esimerkkejä harjoitusaineiston mallin mukaan
- Esim. tekstinluokittelu, sentiment analysis, jopa konekäännös
- Luokat pitää tietää etukäteen
- Koneoppiminen perustuu piirteisiin, eli siihen, miten aineisto esitetään ohjelmalle
- Antaa mahdollisuuden myös luokkien tyypillisten piirteiden tarkastelulle (vrt. avainsana-analyysi)
- “Perinteiseen” avainsana-analyysiin verrattuna monia etuja

# Koneoppimisen kaksi päämenetelmää

## *Ohjaamaton koneoppiminen*

- Ryhmittelee ennalta tuntematonta aineistoa samankaltaisiin ryhmiin
- Ei vaadi etukäteistietoa ryhmien / luokkien määrästä
- Ei vaadi harjoitusaineistoa
- *Mitä aihepiirejä tviitit käsittelevät?*
- *Mitä hymiöitä käytetään samankaltaisissa tilanteissa?*



# Topic modelling

- Ohjaamattoman koneoppimisen muoto
- Tavoite tarkastella aineiston dokumenttien aihepiirejä
  - *Mitä aihepiirejä lehtiartikkelit / tviitit / nettikeskustelut käsittelevät?*
- Topic modelling
  - eristää aineistosta sen topiikit + näitä kuvaavat avainsanat
  - määrittää, mitkä dokumentit sisältävät mitä topiikkeja
  - yksi dokumentti voi sisältää useita topiikkeja
- Muodollisemmin
  - A document is a distribution over topics!!!!
  - Each topic is a distribution over words

## Topics

gene 0.04  
dna 0.02  
genetic 0.01  
...

life 0.02  
evolve 0.01  
organism 0.01  
...

brain 0.04  
neuron 0.02  
nerve 0.01  
...

data 0.02  
number 0.02  
computer 0.01  
...

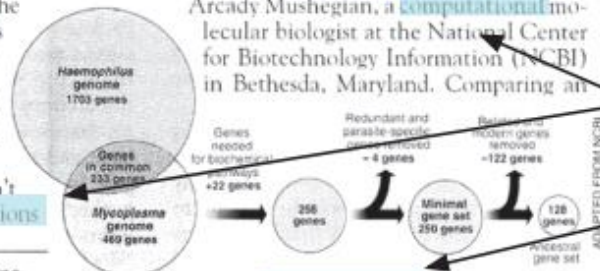
## Documents

### Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to **survive**? Last week at the genome meeting here,\* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those **predictions**

"are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a **genetic numbers game**, particularly as more and more **genomes** are completely mapped and sequenced. "It may be a way of organizing any newly **sequenced genome**," explains Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

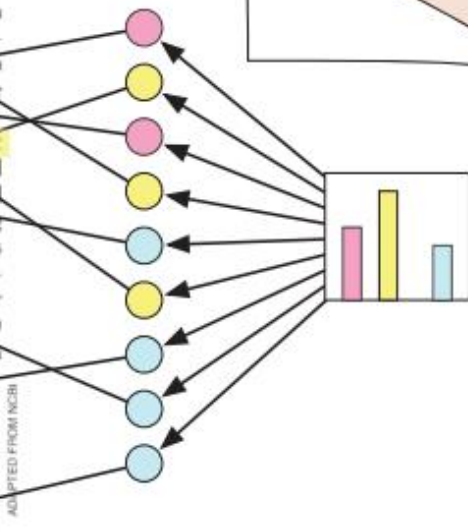


\* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

## Topic proportions and assignments

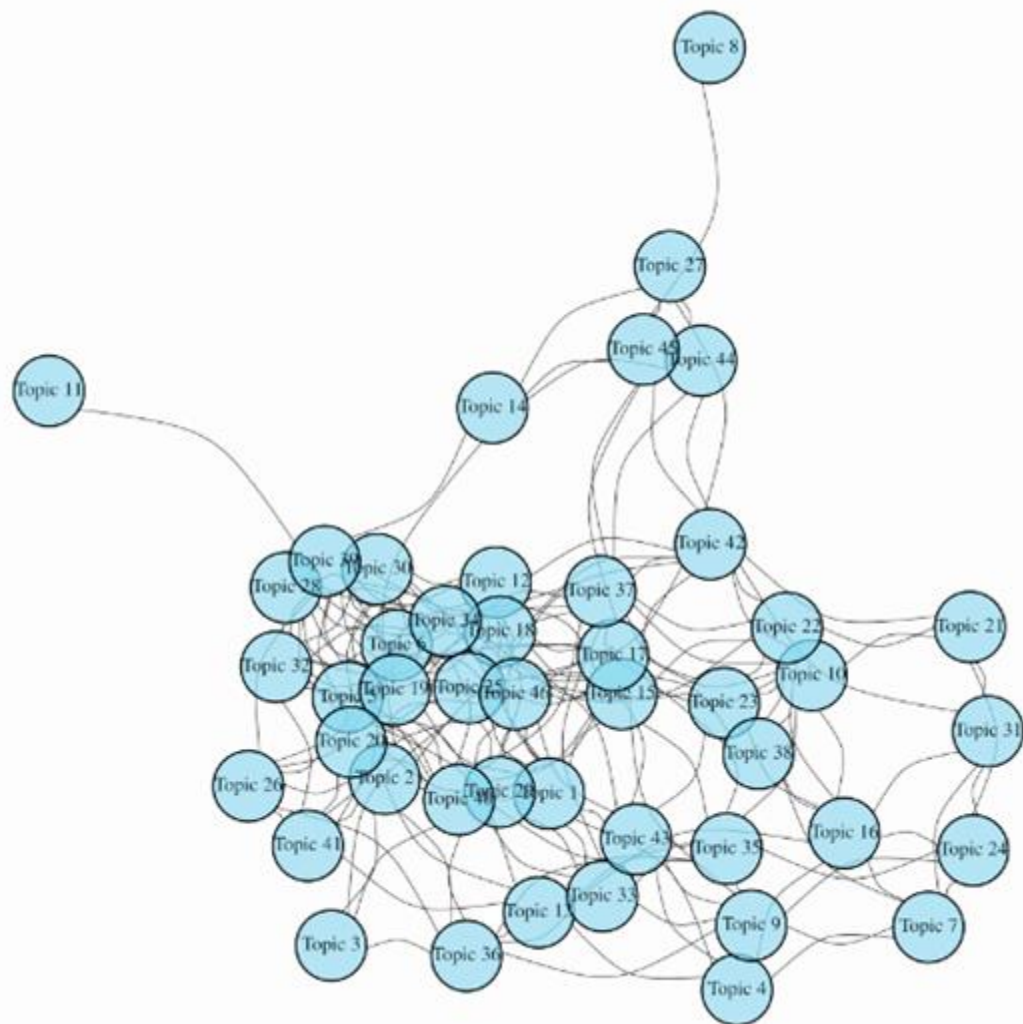


# Miten köyhyydestä keskustellaan Suomi24-palstalla?

- Structural topic modeling (STM), implemented in R
  - STM allows the inclusion of metadata associated with the documents
  - Dataset all Suomi24 comments with the lemma *köyhä* or its near synonym\* from 2014
- 32,407 comments

\* e.g. sossupummi, rutiköyhä, varaton, persaukinen, pienipalkkainen, tyhjätasku...

46 topics



Topic 32: kone, **ajella**, **malli**, tonni, puhelin, tietokone, liikenne, merkki, käyttö, ostaja, **mersu**, **laatu**, ostaja, **audi**, moottori, **skoda**...

Topic 30: **kateellinen**, kateus, järki, henkilö, kuva, huudella, **sääli**, säälittävä, lihava, päätellä, **pummi**, **pelle**, näköjään, puhe, narsisti, **naurettava**, muna, äijä...

Topic 26: **ruoka**, **nälkä**, **leipä**, **kahvi**, **liha**, **alkoholi**, **tupakka**, kala, peruna, viikko, vesi, terveellinen, pullo, maito, kerätä, marja, ruokkia, herkkua, maistua, keittää, makkara...

Topic 28: **paska**, **luuseri**, sentään, **perse**, **hullu**, **homma**, haista, naama, toinen, **kalja**, jauhaa, peli, **kakara**, **sika**, **duuni**, katu, **läski**, **loisia**, **alkoholisti**, vinkua....

Topic 36: **euro**, **asunto**, **vuokra**, **kuukausi**, talo, summa, helsinki, kuu, leipäjono, asumistuki, omakotitalo, kämpppä, vuokra-asunto, säästö, asuntolaina, kk, kerrostalo, sähkö, omistusasunto



# Tänään

- Tavoitteena nähdä, miten tekstinlouhinta toimii
- Koitetaan sekä topic modelingia että tekstinluokittelua (avainsana-analyysiä)
- Aineistona Suomi24 – iso muttei ihan helppo
- Kaikki tapahtuu Unix-serverillä

# Tehtävä 1: Suomi24:n topiikit

- Suomi24-keskusteluketjuja on mallinnettu Topic modelingin avulla
  - Gensim
  - 50 topiikkia
- Tarkastellaan, miten topic modeling –mallin antamat topiikit korreloivat Suomi24:n aihepiirien kanssa
- Katsotaan tyypillisimpiä dokumentteja jostain topiikista. Onko ohjelma toiminut?
- Tarkastellaan, miten topiikit vaihtelevat vuosittain
- Tätä varten pitää käyttää sekä Unixia että Solria
- Yhteys putty-ohjelmalla
- Serverin nimi 0964.kaj.pouta.csc.fi
- Käyttäjätunnus utu-tunnus, salasana [ututunnus]vaihda

# Tehtävä 2: Tekstinluokittelu

- Miten köyhistä puhutaan Suomi24-palstalla? Minkälaisia diskursseja köyhiin liitetään? Korpusavusteinen diskurssianalyysi.
- Tarkastellaan tätä avainsanojen avulla!

# Tehtävä

- Tarkastellaan avainsanojen avulla, mitä diskursseja köyhiin liitetään
- Tätä varten
  - 1) lasketaan avainsanat
  - 2) tarkastellaan niiden käyttöä kommentteissa
  - 3) ryhmitellään avainsanat semanttisiin kenttiin (ks. Jantusen esimerkki)
  - 4) näiden perusteella analysoidaan, mitä diskursseja köyhiin liitetään
- Mitä tarvitaan
  - luokittelija (SVM)
  - köyhä-kommentit + verrokki-aineisto
- Mitä saadaan ulos
  - avainsanat köyhille
  - luokittelijan toimintavarmuus

# Tehtävä 1: Avainsanojen laskeminen ja tarkastelu kommenttiketjuissa

1. Ladataan aineisto ja tarvittavat koodit Githubista käskyllä `git clone https://github.com/TurkuNLP/Digi_menetelmat.git`
2. Tämän jälkeen komennolla `ls` voi tarkastella, mitä hakemistossa on
3. Hakemistossa pitäisi näkyä `Digi_menetelmat` -niminen hakemisto
4. Pääset tähän hakemistoon komennolla `cd Digi_menetelmat`
5. Käskyllä `ls` voi tarkastella, mitä tiedostoja hakemistossa on
6. Tiedostossa `koyha-kommentit-2014.txt.gz` on Suomi24-palstan kommentteja, joissa mainitaan sana köyhä tai sen synonyymi
7. Tiedostossa `no_koyha.txt.gz` on kommentteja, joissa sanaa ei ole mainittu
8. Käskyllä `less [tiedoston nimi]` voi selata tiedostoa (pois pääsee q-kirjainta painamalla)
9. SVM-luokittelija toimii komennolla  
`python preprocess.py | python svm.py`
10. Ohjelma tuottaa ensin analyysin ohjelman toimivuudesta
  - Precision (saanti) kertoo, kuinka suuri osa ohjelman löytämistä kommenteista on oikein
  - Recall (tarkkuus) kertoo, kuinka paljon relevantteja kommentteja kaikista mahdollisista ohjelma löysi
  - f-arvo on näiden yhdistelmä
  - (<https://www.periscopedata.com/blog/precision-recall-and-roc-curves-for-pregnancy-tests>)
11. Ohjelman tuotoksen (eli ne avainsanat) voi printata tiedostoon komennolla  
`python preprocess.py | svm.py >> tuotos.txt`

## Tehtävä 2: Piirteiden vaihtelu, aineiston putsaus ja parhaan luokittelijan valinta

- Avainsanojen ei aina tarvitse olla avainsanoja. Voi käyttää myös esim. lemma-muotoja. Nämä voivat vaikuttaa tuloksiin!
- Lisäksi aineistoa voi esikäsitellä esim. ottamalla tekstistä pois sanaluokat, jotka eivät tuo paljoa kielellistä tietoa (välimerkit, konjunktiot....)
- Kansiosta löytyy myös ohjelma preprocess-lemmas.py. Tämä käyttää saneiden sijasta lemma-muotoja koyha-kommentit-2014.txt.gz-tiedostosta. Koita, miten tämä vaikuttaa luokittelijan toimivuuteen ja itse avainsanoihin!
- `python preprocess-lemmas.py | svm.py >> tuotos.txt`
- Lisäksi voit koittaa poistaa analyysistä sanaluokkia, joita edustavat sanat eivät tuo juuri kielellistä lisäarvoa. Näitä on ainakin apuverbit (AUX) ja välimerkit (PUNCT).
- Sanaluokkien tageja voi tarkastella komennolla `less koyha-kommentit-2014.txt.gz`.
- Tageja voi lisätä preprocess.py tai preprocess-lemmas.py koodin kohtaan `pos_not_to_keep = [u"PUNCT", u"AUX"]`
- Tiedoston saa auki esim. komennolla `nano koyha-kommentit-2014.txt.gz`
- Koita eri versioita (lemmat, juoksevat sanat, välimerkit yms pois...)
- Valitse versio, joka toimii parhaiten

# Tehtävä 3: lopulliset avainsanat

1. Huom! SVM helposti ylisovittaa, eli valitsee sanoja, jotka esiintyvät harvoin, mutta jotka esiintyessään ovat hyvin merkityksellisiä. Tätä varten yllä oleva luokittelija kannattaa ajaa esim. 10 kertaa.
2. Eli toista `python preprocess-lemmas.py | svm.py >> tuotos.txt` 10 kertaa
3. Tehdään avainsanoista frekvenssilista, jonka perusteella voidaan ottaa lopulliseen tarkasteluun ainoastaan ne, jotka esiintyvät usein lähes kaikissa ajoissa. Frekvenssilista tehdään komennolla  
`cat tuotos.txt | egrep "^[a-z]" | egrep -v "avg" | sort | uniq -c | sort -rn | less`
4. Avainsanojen käyttöä kommentteissa voi tarkastella komennolla  
`zcat koyha-kommentit-2014.txt.gz | egrep -B 30 -A 30 "avainsana" | less`
5. Miltä frekvenssilista näyttää? Valittiinko kaikki sanat joka kerta? Jos ei, tarkastele ensin avainsanoja, joita ei valittu joka kerta. Onko niissä jotain omituista?
6. Lopuksi ota tarkasteluun avainsanat, jotka esiintyvät vähintään 8 kertaa kymmenestä ajosta.

# Tehtävä 4: Avainsanojen tarkastelu

- Tarkastellaan, mitä merkityskenttiä avainsanoista nousee
  - Tämä tapahtuu 1) ryhmittelemällä sanat ensin merkityskenttien mukaan käsin. (Ks. Jarmo Jantusen esimerkki) ja 2) tarkastelemalla avainsanoja kontekstissa, eli lukemalla kommentteja, joissa niitä on käytetty
- Mita semanttisia kenttiä muodostuu? Mistä ne kertovat, mitä diskursseja köyhyyteen tai köyhiin liitetään?
- Ota tulokset muistiin niin, että voit selittää ne muille.



# Ryhmätyö: korpusavusteinen diskurssianalyysi

- Valitkaa jokin käsite tai sana, johon liittyviä diskursseja haluatte tarkastella
- Teen aineiston teille
- Vastaava analyysi kuin köyhille
- Ryhmätyönä esitys työstä
- Esitys sovittuna aikana

# Bonus

- Solr\_data.py etsii Solrista kommentteja. Sitä voi itsekin modifioida
  - Mitä etsitään?
  - Kuinka paljon ladataan?
  - Mihin kansioon data tallennetaan?
  - Python3 solr\_data.py
- Grep\_json.py suodattaa edellisen ohjelman tallentamasta datasta helpommin luettavaa tekstiä
  - Python3 grep\_json.py
  - Lisäksi siihen voi helposti lisätä, miltä vuodelta kommentit haluaa!
- Näiden avulla voi tehdä vaikka mitä aineistoja, joiden eroja voi tarkastella...