NLLR

Määrittelydokumentti

Leo Leppänen

20. tammikuuta 2014

1 Ongelman määrittely

Tämän harjoitustyön tarkoitus on tuottaa Java-ohjelma, joka kykenee ajoittamaan automaattisesti sille syötettyjä Reuters-uutispalvelun uutisia erikseen määriteltyihin aikaikkunoihin.

Ajoitusmetodina toimii ajoitettavan dokumentin temporaalisen kielimallin ja käytettävissä olevan referenssikorpuksen aikapartitioiden kielimallien tilastollinen vertailu, joka pyrkii löytämään referenssikorpuksen aikapartitioiden joukosta sen partition, jonka temporaalinen kielimalli vastaa parhaiten dokumentin temporaalista kielimallia.

Referenssikorpuksena toimii eräs Reuters-korpuksen[1] versio.

2 Käytetettävät algoritmit

Harjoitustyössä toteutetaan ensisijaisesti algoritmi nimeltä NLLR, eli Normalized Logarithmic Likelihood Ratio. NLLR on johdettu tavallisesta logaritmisesta uskottavuusosamäärästä[2]. NLLR:n tässä työssä käytetty versio,

[1] http://about.reuters.com/researchandstandards/corpus/

[2] http://fi.wikipedia.org/wiki/Uskottavuusosamäärä

sekä tämän työn inspiraationlähde muutoinkin on de Jong et al. tutkimus Temporal Language Models for the Disclosure of Historical Text[3].

En ole löytänyt vielä selaamastani tutkimuskirjallisuudesta $\mathcal{O}$-analyysiä NLLR-algoritmin toiminnasta. Tarkoitus on siis toteuttaa algoritmi niin nopeana kuin mahdollista ja sen jälkeen analysoida sen aika- ja tilavaativuuksia.

Lisäksi harjoitustyössä käytetään ns. *tf-idf*[4] tilastomenetelmää määrittämään kutakin dokumenttia parhaiten edustavat termit, joille yllämainittua NLLR-algoritmia sovelletaan.

Tf-idfn erään viritellyn implementaation aikavaativuudeksi mainitaan $\mathcal{O}(B \cdot V)$, missä B on keskimääräinen todennäköisyys tarkasteltavan termin esiintymiselle muissa dokumenteissa ja V on sanaston koko. Tämä implementaatio on kuitenkin ilmeisesti varsin haastava ja onkin mahdollista että tässä projektissa tuotettu implementaatio jää aikavaativuuteen $\mathcal{O}(V \cdot D)$, missä V on uniikkien sanojen määrä ja D on dokumenttien määrä.

3 Työn oheistuotteet

Harjoitustyön oheistuotteina syntyy python-skripti joka vähintään muuntaa Reuters-korpuksen sgml-muotoisista tiedostoista Java-ohjelman käsiteltävissä oleviin csv-tiedostoihin. Lisäksi joko ym. python-skripti (käyttäen Natural Language Toolkittiä[5]) tai tuotettava Java-ohjelma (käyttäen todennäköisesti Stanford NLP ohjelmistoa[6]) suorittaa Reuters-korpuksen sisällön typistämisen (Eng. *stemming*).

Nämä oheistuotteet eivät ole osa palautettavaa työtä, eikä niitä ole tarkoitettu arvosteltaviksi.

[3] http://doc.utwente.nl/66448/1/db-utwente-433BCEA2.pdf
[4] http://en.wikipedia.org/wiki/Tf-idf
[5] http://nltk.org/
[6] http://www-nlp.stanford.edu/software/index.shtml