Institutionen för datavetenskap

Department of Computer and Information Science

Examensarbete

Implementation och utvärdering av termlänkare i Java

av

Robin Axelsson

LIU-IDA/LITH-EX-G--13/005--SE

2013-03-22

Linköpings universitet

Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings universitet
581 83 Linköping

Institutionen för datavetenskap

Department of Computer and Information Science

Examensarbete

Implementation och utvärdering av termlänkare i Java

av

Robin Axelsson

LIU-IDA/LITH-EX-G--13/005--SE

2013-03-22

Handledare: Mikael Andersson
Examinator: Magnus Merkel

Abstract

Aligning parallell terms in a parallell corpus can be done by aligning all words and phrases in the corpus and then performing term extraction on the aligned set of word pairs. Alternatively, term extraction in the source and target text can be made separately and then the resulting term candidates can be aligned, forming aligned parallell terms. This thesis describes an implementation of a word aligner that is applied on extracted term candidates in both the source and the target texts. The term aligner uses statistical measures, the tool Giza++ and heuristics in the search for alignments. The evaluation reveals that the best results are obtained when the term alignment relies heavily on the Giza++ tool and Levenshtein heuristic.

Sammanfattning

I detta arbete undersöks möjligheterna att länka samman termer mellan två olika språk. Grundförutsättningen är att det finns en parallell korpus, där meningar är parallellställda på två språk, mellan original och översättning. Förutom den parallella korpusen används indata i form av termkandidater som extraherats i originaltexten och i översättningen. Det här examensarbetet beskriver en implementation av en termlänkare som identifierar vilka termer i källtexten som motsvaras av termer i måltexten. Termlänkaren använder sig av både statistiska mått, verktyget Giza++ och heuristiker i sökningen av länkningar. Utvärderingen avslöjar att det bästa resultatet fås då termlänkningen förlitar sig på Giza++-verktyget och Levenshtein-heuristiken.

Innehåll

1 Inledning

1.1 Syfte

Syftet med det här examensarbetet är att undersöka termlänkning i samband med termextraktion där termsökningen görs innan termlänkningen. En termlänkare ska implementeras och utvärderas. Termlänkaren ska använda sig av statistik och heuristiker i sökningen av länkar.

1.2 Metod

Valet av texter som används i denna studie gjordes innan arbetet startade och ingick i ett tidigare pågående projekt på Linköpings universitet (LiU). De texter som används är patenttexter som kommer från Patentverket och en programvarumanual. De engelska patenttexterna fanns tidigare i digital form och kunde enkelt sparas ner och användas. De svenska patenttexterna var tvungna att skannas in med ett OCR-program vilket har lett till vissa teckenfel. Texterna har länkats på meningsnivå. Resultatet av den länkningen är två xml-filer, en för vardera språket (svenska och engelska). Därefter används programmet iPhraxtor på xml-filerna för att extrahera respektive texters termer (Merkel, Foo & Ahrenberg, 2013). Resultatet från iPhraxtor är två termfiler som innehåller länkningar till respektive texters termer.

För att kunna skapa en termlänkningsapplikation användes artiklar och böcker som källa för att bygga upp en teoretisk bas inom områdena termextraktion och ordlänkning. Dokumentationen för Java (Oracle, 2012) och Perl (Online Perl Documentation, 2012) användes som hjälp för implementationen. Giza++-programmet används till att generera statistiska data som behövs samt förslag på länkningar. De här resurserna användes sedan för att fatta beslut om en potentiell länkning. Perl användes för diverse skript med Giza++ som till exempel sortering av sannolikhetslexikonet, extrahering av den råa texten ur xml filerna osv. Med den teoretiska basen och dokumentationen implementerades termlänkarapplikationen i Java. Applikationens utdata är i form av en termfil som sedan användes i utvärderingen. Utvärderingen gjordes med IEval som är ett verktyg som beräknar prestandamått givet ett facit och en termfil som utvärderas. Facit skapades manuellt av Mikael Andersson (handledare) och Magnus Merkel (examinator).

1.3 Rapportens uppbyggnad

Rapporten är uppdelad i fyra kapitel. Kapitel 1 är en inledning till arbetet och definierar syftet med arbetet och beskriver metodiken för hur arbetet har utförts. Kapitel 2 beskriver den teoretiska bakgrund som används i arbetet. Där begrepp som termextraktion, statistisk maskinöversättning och utvärderingsmått definieras och beskrivs. Kapitel 3 ger en överblick av implementationen av termlänkaren och presenterar utvärderings resultaten. Kapitel 4 är en diskussion om resultaten och framtida förbättringar som går att göra. En slutsats för arbetet presenteras.

1.4 Termer

I rapporten används en del termer och förkortningar som kan vara nya för läsare.
Här nedan följer en lista på förkortningar.

Förkortning	*Utskrivning*	*Förklaring*
SMT	Statistical Machine translation	maskinöversättning via statistik
XML	eXtensible Markup Language	märkspråk i textformat

Nedan följer en ordlista.

Output utdata.

Korpus en samling språkliga data, oftast längre texter.

Stokastisk modell matematisk modell där ett upprepat skeende eller fenomen tillåts ha olika förlopp, utan någon bestämd orsak.

Diskriminerande modell en modell som estimerar den betingade sannolikheten.

Tokenisering att dela upp en text i tokens, d.v.s. löpord.

Heuristik smarta gissningar som hjälper till att hitta lösningar till ett problem.

Bitext parallella texter, t.ex. ett original och dess översättning.

Samförekomst när ett ord förekommer i en mening och ett annat ord förekommer i dess parallella mening sägs de två orden samförekomma.

2 Teoretisk bakgrund

Det här kapitlet beskriver den teoretiska bakgrund som används för rapporten. Termextraktion, statistisk maskinöversättning och utvärderingsmått definieras och beskrivs.

2.1 Termextraktion

Termextraktion kan enkelt sägas vara processen att extrahera termer från en korpus. För att förstå den här processen måste vi först definiera vad en term är.

En term är en sekvens av ett eller flera ord som representerar ett begrepp (Ahrenberg, 2009). Begrepp beskriver egenskaper av objekt i världen. Det finns två typer av objekt, vissa är konkreta (t.ex. träd, hus), andra är abstrakta (t.ex. känsla, dröm). Objekt har egenskaper som definierar dem och det är de egenskaper ett begrepp kombinerar för att tillhöra ett objekt. Ett begrepp är antingen individuellt eller generellt, ett individuellt begrepp är enbart anslutet till ett objekt och har väldigt specifika egenskaper (t.ex. *kinesiska muren*), generella begrepp innehåller egenskaper som är väldigt vanliga och generella (t.ex. *mur)* och är därför anslutet till flera objekt. Begrepp har relationer till andra begrepp och bildar begreppnätverk (Suonuuti, 2001).

De flesta termerna är subjektiv och i grundform men termer av typen plural substantiv, verb och adjektiv förekommer också. Den term som väljs för ett begrepp ska vara lingvistiskt korrekt och följa normerna för språket ifråga. I de fall där flera termer används för att beskriva ett begrepp så är det rekommenderat att använda enbart en term för att representera ett begrepp.

Termer är alltså det språkliga uttrycket för ett begrepp, vilket ser ut att innebära att vi måste veta vad ett begrepp är och ha kunskapen att veta om ett visst textsegment representerar ett begrepp eller ej för att kunna extrahera termer. Enligt Jacquemin & Borigault (2003) så föredras en mer pragmatisk metod än att använda sig av begrepp då vi sällan har tillgång till begrepp. Termextraktion är en process där man extraherar termer från en text, en överblick av den här processen kan ses i figur 1. Extraktion kan göras automatiskt eller manuellt, i många fall följs en automatisk extraktion av en manuell fas för validering av de termer den automatiska processen har extraherat. Termerna som vi får ut av den automatiska extraktionen brukar kallas *termkandidater.* De termer som går igenom valideringen upphöjs sedan till termstatus (Ahrenberg, 2009, s. 2).

Figur 1. Fyra moduler i termextraktionsprocessen. Källa: (Ahrenberg, 2009)

När man söker efter termkandidater kan ett ords lingvistiska egenskaper utnyttjas. Termer är oftast nominalfraser och därför fokuserar termextraktion mest på att hitta just nominalfraser. För en taggad korpus definerar Justeson & Katz (1995) engelska termer som:

”((Adj|Noun)+|((Adj|Noun)*(Noun|Prep)?)(Adj|Noun)*)Noun”

Exempel på termer det reguljära uttrycket hittar:

- solid/Adj fat/Adj content/Noun
- consisting/Noun of/Prep triglycerides/Noun
- potassium/Noun sorbate/Noun

Reguljära uttryck hjälper till att hitta termer men de hittar även många ord som ej är termer trots att de är adjektiv eller substantiv. Sådana ord kan samlas i *stopplistor* som hjälper till att sortera bort icke-termer vid termextraktionen (Ahrenberg, 2009).

2.1.1 Termers statistiska egenskaper

Frekvensen för hur många gånger ett ord förekommer i en korpus är en statistisk egenskap som används. Flera ords frekvenser kan kombineras för att beräkna samförekomstmått för ord. Vi skriver sannolikheten för samförekomst av två ord s_n och t_m som $p(s_n, t_m)$. Samförekomstmått används för att få en bild av hur ofta ord förekommer tillsammans som flerordsenheter i dokument, och för att uppskatta sannolikheten för att ord på olika sidor av en tvåspråkig korpus är varandras översättning (Ahrenberg, 2009).

2.2 Statistisk maskinöversättning

Giza++ är ett opensource-verktyg som använder IBM’s översättningsmodeller 1-5. Verktyget genererar en tvåspråkig samförekomst-fil, där varje källspråksord eller term är listat med dess möjliga översättningar och sannolikheten för översättningen. Genom att sortera ett källspråksord eller term baserat på sannolikheten hittas den mest sannolika översättningen. Verktyget producerar även en länkfil där för varje par av källspråksmening och målspråksmening ger en lista med översättningar av källspråksorden till målspråksorden (Josef, 2000).

2.2.1 Generativa länkningmodeller

Generativa länkningsmodeller använder sig av statistisk maskinöversättning *(eng. SMT – statistical machine translation)* (Tiedemann, 2011, s.60-62).
I SMT försöker man bygga en stokastisk modell som sedan används för att översätta från ett språk till ett annat. Översättningen görs genom att söka efter den mest sannolika översättningen för en given mening enligt den stokastiska modellen. För att uppnå detta innehåller alla SMT-system en översättningsmodell som modellerar sannolikheten för att målsträng **t** är översättningen av källsträng **s**, **P(t|s)**. Då meningar översätts en i taget behövs parallella korpusar som är sorterade på meningsnivå och uppdelade i tokens för att träna SMTn (Tiedemann, 2011, s.60-62).

Meningarna delas upp i mindre beståndsdelar bestående av ord och skiljetecken, även kallade tokens. Detta görs genom en tokenisering som delar upp meningen på ett naturligt sätt. Med det kan vi säga att källmening **s** består av en sekvens av N tokens (s1,...,sN), och att målmening **t** består av en sekvens av M tokens (t1,...,tM). Om ett token är en översättning till ett annat token så säger vi att de är länkade.
Denna länkning betecknas med variabeln **a.** Med mål- och källmening utgör detta en stokastisk länkningsmodell **P(t, a|s)**. Med länknings-modellen kan vi estimera översättningsmodellen som summan av alla möjliga länkningar och deras sannolikheter enligt länkningsmodellen

$$P(t|s) = \sum_{a} P(t, a|s)$$

På samma sätt kan vi även antyda den bästa länkningen â enligt modellen

$$\hat{a} = argmax\, P(t, a|s)$$

Detta betyder att vi kan ta fram en länkning på tokennivå för alla par av meningar när vi har en översättningsmodell för det språkparet. Länkning kan ses som en biprodukt av statistisk översättningsmodellering (Tiedemann, 2011, s.60-62).

2.2.2 Diskriminerande anpassningsmodeller

I diskriminerande modeller så är det viktigt med lämpliga funktioner som hittar diskriminerande egenskaper i den givna datamängden. Funktioner som eftersöks ska vara generella på så sätt att små mängder av träningsdata är tillräckligt för att hitta relationer mellan inputmängden och outputmängden. En nackdel med funktioner är att det finns en risk att man har funktioner som missleder träningsproceduren. Funktioner kan delas upp i olika kategorier beroende på vad de använder och behöver för data. Vi kan skilja mellan lokala funktioner, funktioner som bygger på historik och globala funktioner. Lokala länkningsfunktioner undersöker egenskaper i sammanhang av tokenpar (sn, tm) för att bestämma om dessa tokens ska länkas eller inte (Tiedemann, 2011, s.81-88).
Sådana funkioner definieras som

$$h(a_{nm}, n, m, s, t)$$

Där **n** och **m** refererar till positionen inom deras respektive mening **s** och **t**. En viktig egenskap som kan användas av funktioner är samförekomst av tokens. Samförekomst för tokens fås genom att träna på statistik från parallella korpusar. Det finns många standardmått som använder sig av samförekomstegenskapen, Dice-koefficienten och t-score är två exempel. Vi antar att tokens är enskilda händelser och att sannolikheten för att de förekommer i en parallell text är estimerad från deras relativa frekvenser. Sannolikheten för deras samförekomst kan då definieras som:

$$p(s_n, t_m) \approx \frac{C_B(s_n, t_m)}{|B|}$$

där $C_B(s_n, t_m)$ är antalet meningspar där båda tokens förekommer och $|B|$ är antalet meningspar i den parallella texten. Sannolikheten för förekomst av ett enskilt token kan defineras på liknande sätt.

$$p(s_n) = \sum_{t'} p(s_n, t') \approx \frac{C_B(s_n, \bullet)}{|B|}$$

där $C_B(s_n, \bullet)$ representerar antalet källmeningar som innehåller ordet eller termen s_n, $p(t_m)$ defineras på samma sätt med skillnaden att man tar hänsyn till antalet målmeningar som innehåller ordet eller termen. Med de här definitionerna kan vi beräkna de betingade sannolikheterna (Tiedemann, 2011, s.81-88).

$$p(t_m|s_n) = \frac{p(s_n, t_m)}{p(s_n)} \approx \frac{C_B(s_n, t_m)}{C_B(s_n, \bullet)}$$

De olika samförekomstmått som finns kombinerar de här sannolikheterna baserat på informationsteori för att motivera en potentiell länkning. T-score antar att om sannolikheten för $p(s_n, t_m)$ är identisk eller inom en viss gräns för sannolikheten av slumpmässig samförekomst $p(s_n)p(t_m)$ så finns det ingen stark relation mellan dessa två tokens. Detta antagande formar måttet T-score. (Tiedemann, 2011, s.87)

$$T-score = \frac{p(s_n, t_m) - p(s_n) * p(t_m)}{\sqrt{\frac{p(s_n, t_m)}{|N|}}}$$

Dice-koefficienten är ett mått som letar efter likheter över en viss uppsättning.

$$Dice = \frac{2 * p(s_n, t_m)}{p(s_n) + p(t_m)}$$

Det är vanligt att använda tokenfrekvens för att estimera $p(s_n) \approx freq(s_n)/|B|$, samma som för $p(t_m)$. Detta leder till att värdet som vi får från Dice-måttet kan bli större än 1 då s_n och t_m kan förekomma flera gånger i en och samma mening. Det finns även funktioner som använder andra egenskaper än just statistik. En egenskap är att använda positionen för ett token. Med positionerna för två tokens i deras respektive meningar kan den relativa positionen beräknas som.

$$distance(s_n, t_m) = \lfloor \frac{n}{N} - \frac{m}{M} \rfloor$$

där n och m representerar teckenpositionen i deras respektive meningar, N och M är storleken på respektive meningar i antal tokens.

Den relativa positionen kan sedan användas för att fatta ett beslut om länkning eller användas tillsammans med andra funktioner. Man kan även kolla på ortografiska egenskaper, t.ex. stränglikhet mellan två tokens (Tiedemann, 2011, s.87).

Levenshtein-avståndet är en funktion som beräknar hur många operationer som behöver göras för att göra om en sträng till en annan. Operationerna som görs är substitution, insättning och radering av tecken. Stränglikhet kan ge väldigt varierande resultat beroende på vilka språk man jämför mellan. Korta strängar kan ge låga avstånd just på grund av dess korthet (Jurafsky & Martin, 2000, s.150-156).

2.3 Utvärdering

Det finns två sorters utvärdering som kan användas, extrinsisk (eng extrinsic) och intrinsisk (eng. intrinsic) utvärdering. Extrinsisk utvärdering fokuserar på att mäta prestandan av termlänkningsapplikationen som är baserad på parallella korpusar där prestandan kan till exempel vara hur bra en översättning blir när termlänkningsresultatet används av ett statistisk maskinöversättnings-program (SMT).
Detta kan vara bra då själva termlänkningen ofta bara är en del av applikationen och träningsfasen. Intrinsisk utvärdering kollar istället på individuella länkningar och hur de stämmer överens med ett facit (eng. gold standard). Ett facit är en komplett länkning av tokens som görs manuellt av en eller flera människor. Facit används sedan för att ta fram två prestandamått: precision och recall. Vi har två uppsättningar länkar, L_{gold} som representerar länkingar från facit, och L som är de länkningar som ska utvärderas. (Tiedemann, 2011, s.21-22)

$$Precision = \frac{|L \cap L_{gold}|}{|L|}$$

$$Recall = \frac{|L \cap L_{gold}|}{|L_{gold}|}$$

3 Resultat

Det här kapitlet beskriver de artefakter som används som indata till termlänkaren. En överblick av implementationen ges och resultaten av utvärderingen presenteras.

3.1 Indata

Termlänkningsprocessen behöver termer på svenska och engelska samt statistik som indata för att kunna ge förslag på länkningar. Termerna extraheras från parallella korpusar, med en text för svenska och en för engelska som antas vara varandras motsvarigheter.

3.1.1 Korpus

Korpusarna kommer i form av xml-filer och innehåller strukturerade data om ord sorterad på meningsnivå. Varje mening i den svenska korpusen har en motsvarande mening i den engelska korpusen. En mening representeras som element ”s” och innehåller ett godtyckligt antal ord som representeras som element ”w”. Värdet av ordelementet är ordet som förekommer i korpusen. Utöver det har ord elementet ett antal attribut som beskriver ordets egenskaper. Attribut:

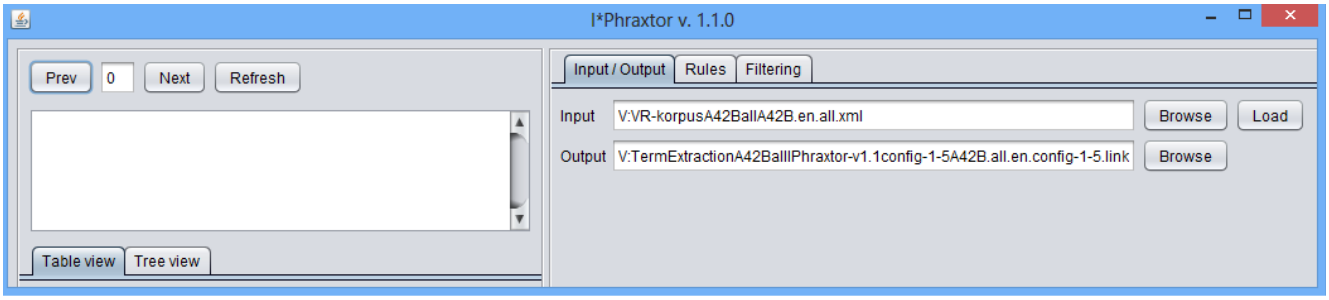
- id → ett unikt id för termen
- relpos → beskriver termens position i meningen
- base → ordet i grundform
- pos → termens ordklass
- msd → böjningar
- func → satsdel
- fa → pekare i analysträdet
- stag → syntaktisk tagg
- sem → semantisk funktion

Ett utdrag på två meningar kan ses i figur 2. Den ena meningen är från den svenska korpusen och den andra meningen är den engelska korpusens motsvarande mening.

```
<s id="s4136" tmxid="FODINA_LT|DOMAIN:A23D|FILE_OR_GROUP:0290065.sv.lwa|ID:167">
    <w id="w81770" sem="" stag="AH" fa="4" func="advl" msd="" pos="ADV" base="isynnerhet" relpos="1">Isynnerhet</w>
    <w id="w81771" sem="" stag="AUX" fa="4" func="v-ch" msd="PAST" pos="V" base="måste" relpos="2">måste</w>
    <w id="w81772" sem="" stag="NH" fa="2" func="subj" msd="PL-NOM" pos="N" base="smakförening" relpos="3">smakföreningarna</w>
    <w id="w81773" sem="" stag="MV" func="main" msd="INF" pos="V" base="vara" relpos="4">vara</w>
    <w id="w81774" sem="" stag=">A" fa="6" func="ad" msd="" pos="ADV" base="åtminstone" relpos="5">åtminstone</w>
    <w id="w81775" sem="" stag="AH" fa="4" func="advl" msd="" pos="ADV" base="delvis" relpos="6">delvis</w>
    <w id="w81776" sem="" stag="NH" fa="4" func="sc" msd="NOM" pos="A" base="löslig" relpos="7">lösliga</w>
    <w id="w81777" sem="" stag="AH" fa="4" func="advl" msd="" pos="PREP" base="i" relpos="8">i</w>
    <w id="w81778" sem="" stag="NH" fa="8" func="pcomp" msd="NOM" pos="N" base="vatten" relpos="9">vatten</w>
    <w id="w81779" sem="" stag="INTERP" func="" msd="Period" pos="INTERP" base="." relpos="10">.</w>
</s>
<s id="s4136" tmxid="FODINA_LT|DOMAIN:A23D|FILE_OR_GROUP:88200666.en.lwa|ID:167">
    <w id="w94163" sem="ADVL" stag="EH" func="meta" msd="" pos="ADV" base="specifically" relpos="1" fa="7">Specifically</w>
    <w id="w94164" sem="" stag="INTERP" func="" msd="Comma" pos="INTERP" base="," relpos="2">,</w>
    <w id="w94165" sem="A>" stag=">N" func="attr" msd="" pos="DET" base="the" relpos="3" fa="4">the</w>
    <w id="w94166" sem="A>" stag=">N" func="attr" msd="NOM-SG" pos="N" base="flavor" relpos="4" fa="5">flavor</w>
    <w id="w94167" sem="SUBJ" stag="NH" func="subj" msd="NOM-PL" pos="N" base="compound" relpos="5" fa="6">compounds</w>
    <w id="w94168" sem="+FAUXV" stag="AUX" func="v-ch" msd="AUXMOD" pos="V" base="must" relpos="6" fa="7">must</w>
    <w id="w94169" sem="-FMAINV" stag="VA" func="main" msd="INF" pos="V" base="be" relpos="7">be</w>
    <w id="w94170" sem="AD-A>" stag="E>" func="ad" msd="" pos="ADV" base="at--least" relpos="8" fa="9">at--least</w>
    <w id="w94171" sem="AD-A>" stag="E>" func="ad" msd="" pos="ADV" base="partially" relpos="9" fa="10">partially</w>
    <w id="w94172" sem="PCOMPL-S" stag="NH" func="comp" msd="ABS" pos="A" base="soluble" relpos="10" fa="7">soluble</w>
    <w id="w94173" sem="ADVL" stag="EH" func="mod" msd="" pos="PREP" base="in" relpos="11" fa="10">in</w>
    <w id="w94174" sem="<P" stag="NH" func="pcomp" msd="NOM-SG" pos="N" base="water" relpos="12" fa="11">water</w>
    <w id="w94175" sem="" stag="INTERP" func="" msd="Period" pos="INTERP" base="." relpos="13">.</w>
</s>
```

Figur 2 Exempel på en parallell mening.

Den svenska och den engelska korpusen används på två sätt. Först fungerar de som data till Giza++-verktyget som genererar en statistisk samförekomst-fil och en länkningsfil. Verktyget kan inte hantera parallella korpusar som xml-filer, därför extraheras omärkta meningar ut från korpusarna med ett Perl-skript som producerar två textfiler, en för vardera språket. Den andra användningen är i termextraktionen. För att extrahera termer från den svenska och den engelska korpusen används verktyget Iphraxtor (Merkel et al, 2013) och kan ses i figur 3. Iphraxtor tar en korpus som xml-fil och genererar en termfil som pekar ut termkandidaterna i korpuset.



Figur 3 Iphraxtor-verktyget.

Två parallella korpusar används för utvärderingen dvs. de parallella korpusarna finns i två versioner. En version innefattar hela korpusens textmängd, den andra är ett urval av hela korpusen och som används för utvärderingen. Versionen som innehåller den fullständiga korpusen används med Giza++-verktyget för att bygga upp statistiska data.
De korpusarna som används för utvärderingen är följande:

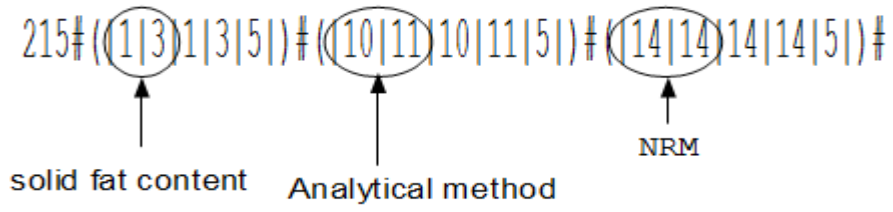
- ”A23D” (Patentansökningar från PRV/EPO) :
  - Utvärderingsversionen:
    - ”A23D.en.test-Robin.xml” engelsk korpus.
      - Filstorlek: 293kB
      - Antal meningar: 100st
      - Antal ord: 2290st
      - Termkandidater: 355st
    - ”A23D.sv.test-Robin.xml” svensk korpus.
      - Filstorlek: 251kB
      - Antal meningar: 100st
      - Antal ord: 1986st
      - Termkandidater: 370st
  - Fullständiga versionen:
    - ”A23D.en.all.xml” engelsk korpus.
      - Filstorlek: 103,3MB
      - Antal meningar: 35403st.
      - Antal ord: 820178st
    - ”A23D.sv.all.xml” svensk korpus.
      - Filstorlek: 87,3MB
      - Antal meningar: 35403st
      - Antal ord: 701385st

- ”acc” (Dokumentation till Microsoft Access 2.0)
    - Utvärderingsversionen:
        - ”acc.en.test.xml” engelsk korpus.
            - Filstorlek: 181kB
            - Antal meningar: 100st
            - Antal ord: 1393st
            - Termkandidater: 273st
        - ”acc.sv.test.xml” svensk korpus.
            - Filstorlek: 157kB
            - Antal meningar: 100st
            - Antal ord: 1232st
            - Termkandidater: 287st
    - Fullständiga versionen:
        - ”acc.en.all.xml” engelsk korpus.
            - Filstorlek: 26,4MB
            - Antal meningar: 14700st
            - Antal ord: 204460st
        - ”acc.sv.all.xml” svensk korpus.
            - Filstorlek: 22,4MB
            - Antal meningar: 14679st.
            - Antal ord: 176239st

Då versionen som innehåller den fullständiga korpusen endast används med Giza++-verktyget och inte Iphraxtor så finns ingen statistik på hur många termkandidater de innehåller.

3.1.2 Termfil

Definitionen av en termfil är en fil som pekar ut termkandidater i en korpus. Iphraxtor-verktyget genererar en termfil per korpus, syntaxen på den termfilen är den syntax som genereras från termlänkaren och används i utvärderingen. Exempel på termkandidater som pekas ut kan ses i figur 4. De två första sifforna inom varje parentes pekar ut en källtermkandidat, tredje och fjärde siffran i varje parentes pekar ut vilken måltermkandidat källtermkandidaten ska länkas mot. Då Iphraxtor-verktyget endast extraherar termkandidater ur en korpus och ej gör någon länkning så pekar termkandidaterna på sig själva.



Figur 4. Exempel på från Iphraxtor termfil. Orden pekar på den källterm de tillhör.

3.1.3 Giza++-länkfil

Giza++-verktyget genererar en länkfil där ord i käll- och måltexterna kopplas samman. Den länkfilen används som stöd för att ta beslut om en potentiell länkning under själva länkningsprocessen. I figur 5 kan ett exempel ses på en mening som har länkats av Giza++. Där man kan se målmeningen (engelska) och hur varje ord i källmeningen länkas mot målord. Varje källord har en lista med vilket eller vilka målord källordet pekar på. T.ex. ”vattenpermeabilitet” pekar på målord 6 och 7, vilket är ”water” och ”permeability”.

```
# Sentence pair (5) source length 6 target length 8
alignment score : 7.94973e-08
it should have a low water permeability .
NULL ({ 4 }) det ({ 1 }) bör ({ 2 }) ha ({ 3 }) låg
({ 5 }) vattenpermeabilitet ({ 6 7 }) . ({ 8 })
```

Figur 5. Exempel på länkning av en mening i Giza++-länkfil.

3.1.4 Samförekomst-fil

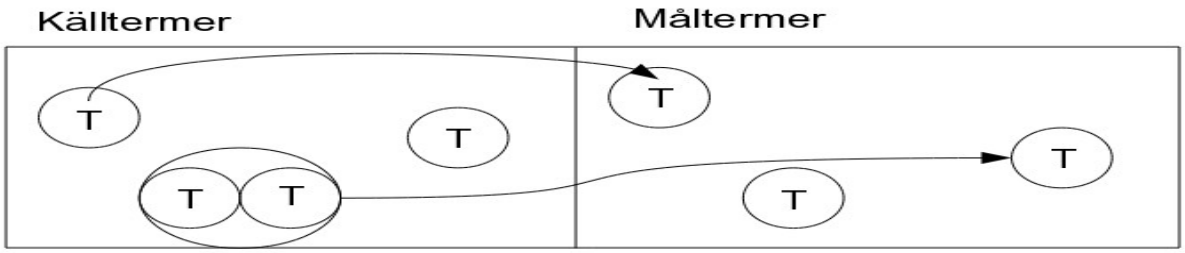
Giza++-verktyget genererar förutom en länkfil även en samförekomst-fil som innehåller statistik för hur ord i käll- och målkorpus samförekommer. Om ett källord förekommer i en källmening och ett målord förekommer i en parallell målmening så samförekommer källord och målord. Samförekomstfilen beskärs för att ta bort samförekomster med låg sannolikhet och även teckenkombinationer som inte är giltiga ord. Filen sorteras på källord för att snabbare kunna hitta flera förekomster av ett visst källord. Ett exempelutdrag från en samförekomst-fil kan ses i tabell 1. Sannolikhets-kolumnen ger en siffra på hur stor sannolikhet det är att källordet översätts som målordet på samma rad.

Källord	**Målord**	**Sannolikhet**
Faster	Snabbare	0.80387
fastest	snabbast	1
Free	Produkter	0.100761
fat	fett	0.325028

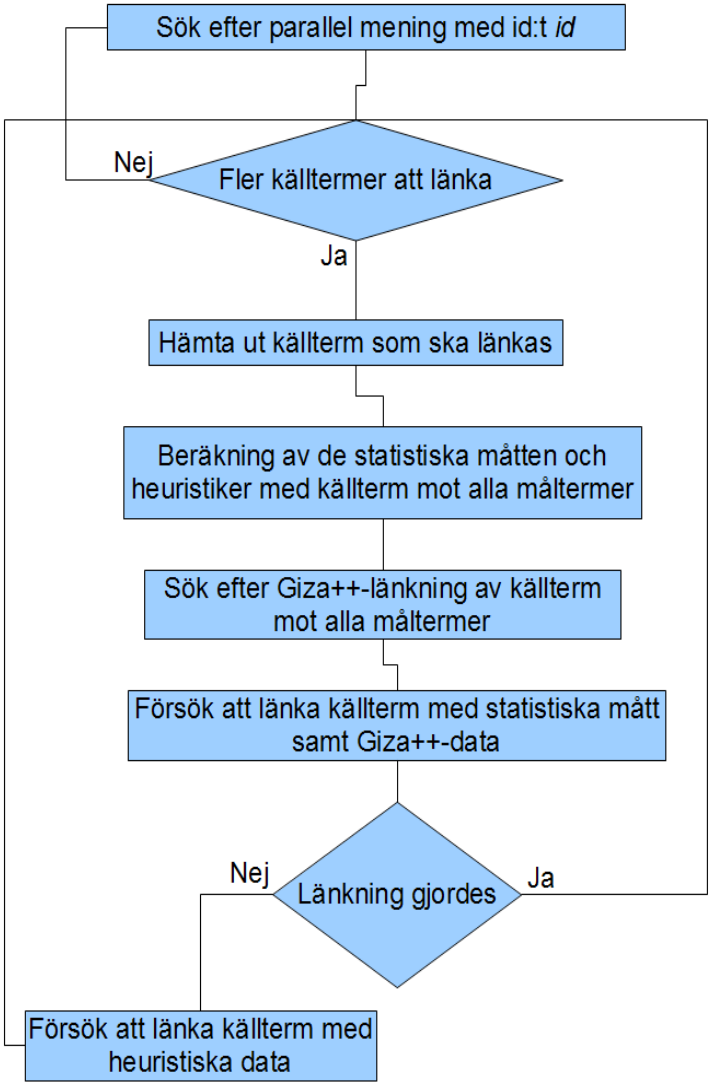
Tabell 1. Exempelutdrag från samförekomst-fil.

3.2 Implementation

Termlänkaren implementerades i Java och tar en parallell korpus, dess två Iphraxtor-termfiler, en samförekomst-fil och en Giza++-länkfil som indata. De länkningar som finns i Iphraxtor-termfilerna pekar på sig själva. Målet är att hitta vad källtermkandidaten eller måltermkandidaten ska länkas mot. Figur 6 illustrerar ett exempel på hur länkningar kan se ut.



Figur 6. Exempel på länkning med två länkningar, där två källtermer har slagits ihop för att kunna länkas mot en målterm.



Figur 7. Process för länkningsalgoritm.

Termerna laddas in till två listor, en som innehåller källmeningar och en för målmeningar. Listorna innehåller enheter av klassen ”Sentence”. Klassen innehåller ett id för meningen samt en lista över de termer som finns i varje ”Sentence”. Hur termlänkningsprocessen ser ut kan ses i figur 7 som visar alla moduler processen består av.

En räknare används för att iterativt gå igenom meningarna. Räknaren ger ett id på vilken mening som eftersöks, termlänkaren söker sedan igenom listorna efter en mening med det idt som räknaren är på. I de tre fallen där inga parallella meningar hittas görs ett test för att hantera tre olika fall.Fallen är:

1. Ingen mening i varke sig käll- eller mållistan hittades.
   Skriv ut enbart meningsnumret i länkfilen, inga länkningar gjorda.

2. Mening hittades i källlistan men ej i mållistan.
   Länka termerna i källmeningen till null.

3. Mening hittades i mållistan men ej i källlistan.
   Länka termerna i målmeningen till null.

I fallet där två parallella meningar hittades itereras källtermerna en efter en och länkningsmöjligheter beräknas mot måltermer med statistiska mått, heuristiker samt Giza++-länkning. De statistiska måtten och Giza++-länkning används som källor för primär länkning och heuristiker som sekundär. Efter beräkning tas ett beslut om vad källtermen ska länkas mot.

För att ta beslut om en länkning används en viktningsstrategi. Om de statistiska måtten eller Giza++-länkningen hittar evidens för en potentiell länkning viktar de den måltermen. Viktningen representeras som en lista innehållandes flyttal. Listan är lika stor som antalet måltermer och vikten för alla måltermer initieras till 0. När viktning av en målterm görs så adderas en vikt för den målterm som ska viktas. Formel 3.1 definerar hur en målterm viktas.

$$(3.1) \quad målterm_{vikt} = målterm_{vikt} + mått_{vikt}$$

Vikten för ett mått är olika beroende på vilket mått det är som viktar och vilken strategi som används. Efter viktningen så initieras en sökning efter den målterm med högst viktning. Om viktningen är över tröskelvärdet tas det som bevis på att en länkning ska göras. Om viktningen är under tröskelvärdet tolkas det som att det inte finns någon evidens för att källtermen ska länkas mot någon av måltermerna. I de fallen där de statistiska måtten samt Giza++ inte hittade evidens nog för en potentiell länkning används heuristikerna. Heuristikerna använder en egen intern viktningsstrategi för att ta beslut om en länkning.

Ihopslagning av termer används på de termer som inte länkas. Figur 8 visar ett exempel på en gynnsam ihopslagning av termer. För att slå ihop två termer krävs att deras relativa position i meningen är i närhet till varandra. Då termer kan bestå av flera ord så beräknas närheten genom att subtrahera den minimala relativa positionen av alla ord i term B med den maximala relativa positionen av alla ord i term A, om närheten beräknas till 1 betyder det att termerna står i närhet till varandra och kan slås ihop. Ihopslagningen görs genom att skapa en ny term och lägga till först de ord som term A består av och sedan de ord term B består av.

$$\begin{aligned} Svensk\ term\ A &= \text{Egenskapen} \\ Svensk\ term\ B &= \text{tillåt redigering} \\ Engelsk\ term\ A &= \text{allowediting form property} \\ Ihopslagen\ svensk\ term &= \text{Egenskapen tillåt redigering} \end{aligned}$$

Figur 8. Exempel på gynnsam ihopslagning av termer.

3.2.1 Statistisk ordlänkning

Statistiken i samförekomst-filen används för att beräkna de statistiska måtten Dice och T-score, men även för att använda sannolikheten för en samförekomst rakt av som benämns som giza-måttet då samförekomst-filen är genererad av Giza++-verktyget. Då en term kan bestå av fler än ett ord så beräknas sannolikheten för att en term samförekommer med en annan term som i formel (3.2).

$$(3.2) \quad p(s_n, t_m) = \prod_N \sum_M Fn(n, m)$$

$$(3.3) \quad (2 * p(n, m)) / p(n) + p(m) \quad \text{- Dice}$$

$$(3.4) \quad (p(n, m) - (p(n) * p(m))) / \sqrt{p(n, m) / |B|} \quad \text{- T-score}$$

Där $p(n,m)$ betecknar den funktion som letar upp sannolikheten för ord n mot ord m i samförekomst-filen. Sannolikheten beräknas genom att först beräkna sannolikheten för källtermen mot måltermen. Det görs genom att summera sannolikheterna källord k har mot alla orden i måltermen. N representerar antal ord i källtermen och M antal ord i måltermen. Sedan multipliceras sannolikheterna och ger sannolikheten för samförekomsten av en term mot en annan term. $Fn(n,m)$ är en funktion som definieras olika beroende på vad det är som ska beräknas.

Funktionen är definerad som formel (3.3) vid beräkning av dice-koefficienten av en term mot en annan, och som formel (3.4) vid beräkning av T-score. För gizamåttet så används $p(n,m)$ -funktionen rakt av. Om ingen sannolikhet hittas i samförekomst-filen ges sannolikheten 0. Sannolikheterna för källtermen mot måltermerna sparas för de olika måtten och efter beräkning av alla sannolikheter hämtas maxsannolikheterna ut och vilka termer det är mot. Maxsannolikheterna itereras och om sannolikheten är över ett visst tröskelvärde så viktas den måltermen som är knuten till sannolikheten. Tröskelvärdet är 1/10 av max värdet, och eftersom måtten har olika maxvärden medför detta att de har olika tröskelvärden. Tröskelvärdena för de olika måtten kan ses i tabell 2.

Mått	**Tröskelvärde**
Gizamåttet	0.1
Dice koefficienten	0.2
T-score	3540

Tabell 2. Tröskelvärden för de statistiska måtten.

Om en maxsannolikhet är över dess tröskelvärde så viktas den termen som potentiell länkning.
Om en maxsannolikhet är under tröskelvärdet så tolkas det som att det inte finns någon evidens som pekar på att källtermen ska länkas med någon av måltermerna och ingen viktning görs.

3.2.2 GIZA++-länkning

För Giza++-länkning används filen A3 som genereras med Giza++. A3 är en länkfil som länkar ord från målmeningen till ord på källmeningen i två parallella meningar i taget. Filen används för att söka om Giza++ har gjort en länkning mellan två termer. Då termer kan bestå av fler än ett ord, kontrolleras det att alla källord länkas till åtminstone ett målord. Om alla källord har en länkning mot något mål ord så har Giza++ en länkning mellan de två termerna och vi viktar måltermen.

3.2.3 Heuristiker

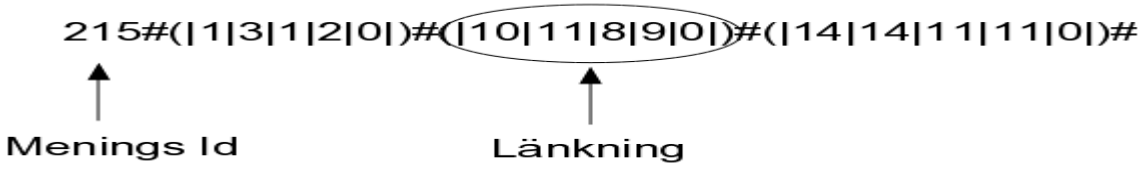
I de fallen där ingen statistik eller Giza++-länkning hjälper för att hitta en länkning för en term används heuristiker som ett sista sätt att försöka hitta en länkning. De heuristiker som används är levenshtein-avståndet (Jurafsky & Martin, 2000, s.150-156) och relativ position i meningen. Levenshtein-avståndet är definierat som det minsta antal operationer på en sträng för att transformera den till en annan sträng. Operationerna som används är lägga till, ta bort eller byta ut ett tecken. Färre antal operationer som behöver göras betyder mindre avstånd mellan två strängar och större chans för att de ska länkas mot varandra. Då termer kan bestå av fler än ett ord så konkateneras orden i källtermen samt måltermen den ska matchas mot till två strängar. De konkatenerade strängarna används som input till levenshtein-heuristiken.

Den andra heuristiken som används är en heuristik som kollar på den relativa positionen för termer. Om källtermens och måltermen vi försöker matcha mot ligger relativt lika positionerade i sina meningar säger vi ha vikt för ett stöd för en potentiell länkning.
För att ta beslut om en potentiell länkning används även här en viktningsstrategi. För levenshtein-avståndet används ett tröskelvärde för att kontrollera att källtermen är tillräckligt lik måltermen, om levenshtein-avståndet är under tröskelvärdet viktas källtermen.

Den relativa positionsheuristiken kontrollerar att positionen är inom ett visst intervall. Om positionen är inom intervallet så viktar den käll termen. Viktningen för heuristikerna fungerar på samma sätt som för de statistiska måtten. Om levenshtein-avståndet eller relativa position heuristiken hittar evidens för en länkning adderar den en vikt på den måltermen som ska viktas. Vikten beror på vilken heuristik och ivilken viktningsstrategi som används. När alla måltermer har viktas inleds en sökning på den målterm med mest vikt. Om vikten är över ett tröskelvärde så skapar man en länk från källterm mot målterm.

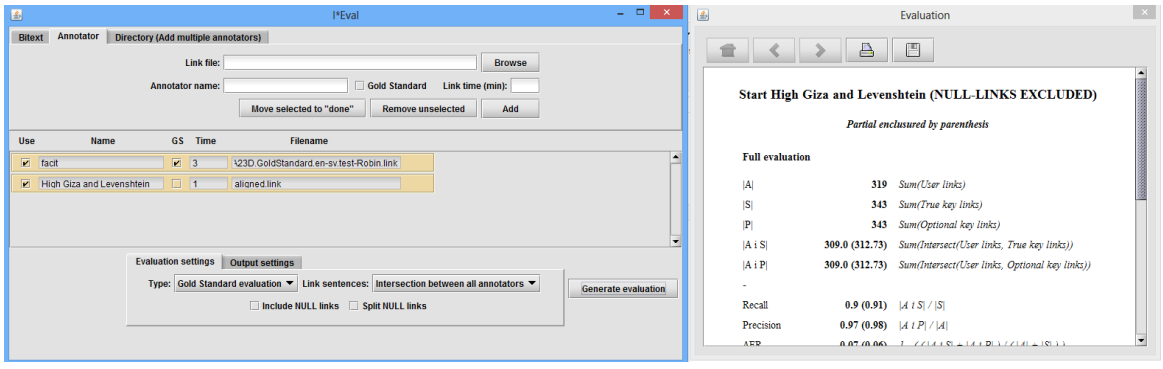
3.3 Utvärdering

Till utvärderingen av länkningar som termlänkaren gör används två vanliga prestandamått, *Recall* och *Precision* (Tiedemann, 2011, s.21-22). Det eftersökta resultatet är ett högt värde på de båda måtten. För utvärderingen används ett verktyg IEval som beräknar recall och precision givet en term-fil som ska utvärderas och en guldstandard, som fungerar som ett facit för hur länkningarna ska vara. Exempel på länkning gjord i en termfil kan ses i figur 9. Där den utpekade länkningen visar att källtermen som består av ord nummer 10 och 11 i källmeningen ska länkas mot måltermen som består av ord nummer 8 och 9 i målmeningen.



Figur 9. Exempel på länkning gjord av termlänkaren.

När ett facit och en term-fil har laddats in i IEval-verktyget kan en utvärdering generas. Figur 10 visar hur IEval-verktyget ser ut. Resultatet av utvärderingen kommer i form av ett html dokument innehållandes recall och precision måtten och kan ses till höger i figur 10.



Figur 10. IEval-verktyget.

Eftersom viktning används för att ta beslut om en potentiell länkning både på den statistiska och heuristiska länkningen gjordes omfattande tester på graden av viktning för de olika måtten. Utgångspunkten för viktning var balanserad dvs. viktningen av de olika delarna påverkar lika mycket. De olika inställningar på viktningen som gjordes är.

- Enbart Giza++-länkning. Endast de länkningar gjorda av Giza++ används, statistiska måtten och heuristikerna används ej.
- Hög viktning av Giza++-länkning. En länkning gjord av Giza++ är viktad tre gånger så mycket som de enskilda statistiska måttens viktning. För heuristikerna användes en balanserad viktning.
- Balanserad viktning av de statistiska måtten och Giza++-länkning. De statistiska måtten och Giza++-länkning viktar lika myket. För heuristikerna används en balanserad viktning.
- Hög viktning av Levenshtein-måttet. Levenshtein-måttet viktar två gånger så mycket som relativa-positions heuristiken. För de statistisk måtten och Giza++-länkning används en balanserad viktning.
- Hög viktning av relativa position-måttet. Relativa position-måttet viktar två gånger så mycket som Levenhstein-avstånds heurstiken. För de statistisk måtten och Giza++-länkning används en balanserad viktning.

Vikten för de olika strategierna kan ses i tabell 3.

Viktningsstrategi	**Vikt av Giza++-länkning**	**Vikt av Dice-koefficienten**	**Vikt av T-score**	**Vikt av Giza-måttet**	**Vikt av Levenhstein-måttet**	**Vikt av relativ position-måttet**
Enbart Giza++-länkning	3.0	0	0	0	0	0
Hög viktning av Giza++-länkning	3.0	1.0	1.0	1.0	1.0	1.0
Balanserad viktning av stat. Mått och Giza++-länkning	1.0	1.0	1.0	1.0	1.0	1.0
Hög viktning av Levenhstein-måttet	1.0	1.0	1.0	1.0	2.0	1.0
Hög viktning av relativ position-måttet	1.0	1.0	1.0	1.0	1.0	2.0
Hög viktning av Giza++-länkning och Levenhstein-måttet	3.0	1.0	1.0	1.0	2.0	1.0

Tabell 3. Viktning av olika länkningsstrategier

3.3.1 Utvärderingsresultat

Länkningsstrategi	Korpus	Recall	Prec.
Enbart Giza++-länkning	A23D	0,85	0,95
Hög viktning av Giza++-länkning	A23D	0,9	0,97
Balanserad viktning av stat. Mått och Giza++-länkning	A23D	0,47	0,98
Hög viktning av Levenhstein-måttet	A23D	0,55	0,94
Hög viktning av relativ position-måttet	A23D	0,48	0,98
Hög viktning av Giza++-länkning och Levenhstein-måttet	A23D	0,91	0,97

Tabell 4. Resultat för A23D korpusen.

Resultaten i tabell 4 visar att den bästa länkningsstrategin för A23D korpusen är hög viktning av Giza++-länkning och Levenhstein-måttet.

Länkningsstrategi	Korpus	Recall	Prec.
Enbart Giza++-länkning	acc	0,54	0,77
Hög viktning av Giza++-länkning	acc	0,67	0,87
Balanserad viktning av stat. Mått och Giza++-länkning	acc	0,55	0,89
Hög viktning av Levenhstein-måttet	acc	0,58	0,85
Hög viktning av relativ position-måttet	acc	0,55	0,86
Hög viktning av Giza++-länkning och Levenhstein-måttet	acc	0,69	0,85

Tabell 5. Resultat för acc korpuset.

Resultaten för acc korpusen kan ses i tabell 5 och visar att två länkningsstrategier är bäst beroende på vad som eftersöks. Hög viktning av Giza++-länkning och hög viktning av Giza++-länkning och Levenhstein-måttet har snarlika resultat där hög viktning av Giza++-länkning har 0,02 mer i precision men 0,02 mindre i recall.

Ett par utdragna exempel på länkningar gjorda med Hög viktning av Giza++-länkning och Levenhstein-måttet kan ses i tabell 6 och 7.

Engelsk mening	"A good indication of expected microbiological keepability can, for--example, be obtained by conductivity measurements."	
Svensk mening	"En bra indikation på förväntad mikrobiologisk hållbarhet kan exemdelvis erhållas genom konduktivitetsmätningar."	
Engelska termer extraherade av Iphraxtor	"microbiological keepability", "conductivity measurements"	
Svenska termer extraherade av Iphraxtor	"mikrobiologisk hållbarhet", "konduktivitetsmätningar"	
Länkningar gjorda av termlänkaren	"microbiological keepability"	"mikrobiologisk hållbarhet"
	"conductivity measurements"	"konduktivitetsmätningar"

Tabell 6. Parallell mening med id 3485 från A23D korpuset.

Engelsk mening	"910-915, Oct. 1984, describe an emulsion consisting of triglycerides composed of medium chain and long chain fatty acids in similar proportions used for sparing body protein in burned rats."	
Svensk mening	"910-915, oktober 1984, beskriver en emulsion bestående av triglycerider, sammansatta av mellankedjiga och långkedjiga fettsyror i likartade proportioner, använda för att spara kroppsprotein hos brända råttor."	
Engelska termer extraherade av Iphraxtor	"emulsion", "triglycerides", "medium chain and long chain fatty acids", "body protein", "burned rats"	
Svenska termer extraherade av Iphraxtor	"emulsion", "triglycerider", "mellankedjiga och långkedjiga fettsyror", "proportioner", "kroppsprotein", "brända råttor"	
Länkningar gjorda av termlänkaren	"emulsion"	"emulsion"
	"triglycerides"	"triglycerider"
	"medium chain and long chain fatty acids"	"mellankedjiga och långkedjiga fettsyror"
	"body protein"	"kroppsprotein"
	"burned rats"	"brända råttor"

Tabell 7. Parallell mening med id 5457 från A23D korpuset.

4 Diskussion

Recall varierar väldigt beroende på korpus och länkningsstrategi, ett lågt recall-värde indikerar att många av termerna inte kommer med i länkningen. För att få användbara resultat krävs inte enbart hög precision utan även hög recall.

Skillnaden i recall mellan hög viktning av Giza++-länkning med balanserade heuristiker och hög viktning av Giza++-länkning med hög levenshtein viktning är minimal och orsaken till att levenshtein inte hjälper till att höja värdet mer beror på att många av de länkningar levenshtein gör är redan gjorda av Giza++. En nackdel med Giza++ är att den (som all statistik) är väldigt beroende av stora mängder data för att kunna göra relevanta länkningar. Med ett litet korpus kommer Giza++ göra många länkningar som är felaktiga och många länkningar mot ingenting. I de fall där man inte har så pass stora datamängder skulle det kunna vara smartare att ha heuristikerna som den primära länkningen och statistiken och Giza++-länkningar som sekundär.

Det bästa recallvärdet för A23D-korpusen är 0,91 och för acc-korpusen 0,69, skillnaden är alltså 0,22 mellan de två korpusarna. Detta kan bero på flera faktorer. Den statistiska länkningen påverkas av att det är mindre data i acc-korpusen. En annan faktor kan vara att termextraktionen inte lyckas extrahera matchande termer från respektive språk.

Ihopslagning av termer används då det kan finnas termer som har splittrats upp som två olika termer i termextraktionen. Den första implementeringen av ihopslagning gjordes på så sätt att efter itereration av alla källtermer så slog man ihop iterativt de källtermer som inte hade länkats mot någon målterm två och två. De ihopslagna termerna gick sedan igenom länkningsalgoritmen. Problemet med denna implementation var dock att den statistiska delen hade ingen chans att lyckas. Detta på grund av att om vi har två termer X och Y så vet vi att enskilt så har de ingen Giza++ länkning (för om de hade det hade de varit länkade) och även att sannolikheten mot alla måltermer är låg, detta vet vi för om sannolikheten hade varit hög hade värdet för de tre statistiska måtten varit över tröskelvärdet och termerna hade haft en länkning. Då vi har två sannolikheter som vi vet är under tröskelvärdena mot alla kvarvarande måltermer så har vi: liten sannolikhet * liten sannolikhet vilket resulterar i en ännu mindre sannolikhet, och därför har ihopslagning av termer ingen chans att lyckas på den statistiska delen. Då heuristikerna inte använder sig av statistik utan andra egenskaper av termer så används ihopslagning bara med heuristikerna.

Anledningen till att jag valde att limitera mig till att bara använda Dice, T-score och giza-måttet var att de är väldigt lätta att förstå och att implementera. Det finns många andra mått såsom loglikehood, mutual information osv. som även de hade kunnat implementerats och använts. Men fler mått gör det svårare och mer komplext att utvärdera och balansera viktnings processen. Om man väljer att använda sig av många mått så kan det vara en bra idé att inte använda en viktningsstrategi.

En alternativ strategi som jag övervägde att använda mig av var en strategi som bygger på prioritetering, det vill säga man evaluerar de mått man har enskilt och för sig och bestämmer sedan en prioriteringsordning, den som är prioriterat som etta är den som är mest pålitlig. Man beräknar det mått som är mest pålitligt först med källtermen mot alla måltermer och om det största beräknade värdet är över tröskelvärdet för det måttet så länkar man källtermen mot måltermen, om inga av värdena är över tröskel värdet så beräknas det mått som prioriterats som tvåa och osv. med de övriga måtten.

4.2 Förbättringar

Det finns en del saker som skulle gå att förbättra i termlänkaren. Att läsa in all statistik och annan data som används i termlänkaren tar lång tid. Termlänkaren använder sig av vissa effektiva operationer i nuläget, såsom att källtermerna är sorterade så när källtermen har hittats så vet applikationen om nästa källterm inte är den samma så har vi redan kollat på alla källtermerna av den vi letar efter och kan avsluta sökningen. Men det finns mycket att effektivisera i applikationen och termlänkningsprocessen som skulle kunna göras i en framtida undersökning.

4.1 Slutsats

Bästa resultatet fås när Giza++-länkningar och Levenshtein-måttet har hög viktning. Precision och recall värdena vi får när de inställningar används är höga och är bra balanserade mot varandra vilket visar att de länkningar som gjorts är i de flesta fall relevanta. De höga värdena kommer till största del från de länkningar Giza++ gjort. Detta visar att Giza++ är en bra bas att bygga vidare på med andra statistiska mått och heuristiker. Måttens och heuristikernas funktion blir då att trycka upp precision och recall de sista procenten som behövs för att resultatet ska bli riktigt bra.

Referenser

Ahrenberg, Lars (1998). *Term extraction: A Review Draft Version 091221*. IDA, Linköpings universitet.

Jacquemin, Christian & Bourigault, Didier (2003). *Termextraction and automatic indexing*'. I: R. Mitkov (red.) *Handbook of Computational Linguistics*, s. 599-615. Oxford University Press, Oxford.

Josef, Franz (2000). *Giza+ Readme* [www] <http://code.google.com/p/gizapp/source/browse/trunk/GIZA%2B%2B-v2/README> Besöktes December 2012.

Jurafsky, Daniel & Martin, James H. (2000) *Speech and Language Processing.* Prentice Hall, 1. uppl.

Justeson, John S. & Katz, Slava M. (1995). *Technical terminology: some linguistic properties and an algorithm for identification in text*. I: *Natural Language Engineering* nr 1. s. 9-27.

Linköpings Universitet [www] <http://www.liu.se/> Besöktes December 2012.

Merkel, M., Foo, J. & Ahrenberg L. (2013). *IPhraxtor - A linguistically informed system for extraction of term candidates.* I: Proceedings of NODALIDA 2013, Oslo.

Online Perl Documentation [www] <http://www.perl.org/docs.html> Besöktes December 2012.

Oracle. *Java Documentation* [www] <http://docs.oracle.com/javase/7/docs/api/> Besöktes December 2012.

Suonuuti, Hedi (2001). *Guide to Terminology*. Nordterm

Tiedemann, Jörg (2011). *Bitext Alignment*. Morgan & Claypool publishers (Syntehesis lectures on human language technologies, lecture 14).

På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Robin Axelsson