UPPSALA
UNIVERSITET

Automatisk morfosyntaktisk analys av kliniska texter

Filip Antomonov

Institutionen för lingvistik och filologi
Språkteknologiprogrammet
Kandidatuppsats i språkteknologi
4 september 2014

Handledare:
Beáta Megyesi, Uppsala Universitet

Sammandrag

Elektroniska patientjournaler har egna lingvistiska egenskaper, vilka har visat sig vara annorlunda från standardspråket. På grund av detta har datorlingvistiska verktyg, som tränats på standardspråket, inte kunnat nå samma prestandanivåer vid bearbetning av klinisk data. I denna kandidatuppsats beskrivs en kedja av verktyg för automatisk bearbetning av svenska kliniska texter, med start i tokenisering och ordklasstaggning, och vidare mot dependensparsning. Utvärderingen av kedjans komponenter visar att gällande verktyg för bearbetning av naturligt språk kan användas, men att prestandan sjunker kraftigt när modeller, tränade på vanligt språk, tillämpas på klinisk data. Kandidatuppsatsen framlägger även en mindre syntaktiskt annoterad datamängd av kliniska texter, ämnad att fungera som guldstandard.

Аннотация

Электронные медкарты имеют собственные лингвистические свойства, отличающиеся от обычного языка. Вследствие этого, инструменты для вычислительной лингвистики, обученные для работы с обычным языком, не достигают того же уровня точности при обработке клинических данных. В этой дипломной работе описывается цепочка инструментов для автоматической обработки шведских клинических текстов, начиная с токенизации и частеречной разметки, и заканчивая парсингом зависимостей. Оценка компонентов цепочки показывает, что имеющиеся инструменты для обработки естественного языка могут использоваться, но при этом наблюдается существенное падение производительности, когда обученные на обычном языке модели применяются на клинические данные. Дипломная работа также представляет небольшой, синтаксически размеченный массив данных из клинических текстов, выполняющий роль золотого стандарта.

Abstract

Electronical health records have their own linguistic characteristics and have been shown to deviate from standard language. Therefore, computational linguistics tools, trained on standard language, do not achieve the same accuracy when applied to clinical data. This thesis describes a pipeline of tools for the automatic processing of Swedish clinical texts from tokenization, through part-of-speech tagging and dependency parsing. The evaluation of the components of the pipeline shows that existing tools for natural language processing can be used, but performance drops greatly when models trained on standard language are applied to clinical data. The thesis also presents a small, syntactically annotated data set of clinical text to serve as gold standard.

Innehåll

Sammandrag **2**

Förord **4**

1 Inledning **5**
1.1 Syfte 6
1.2 Upplägg 7

2 Bakgrund **8**
2.1 Kliniska texter 8
2.2 Verktyg 9

3 Metod **14**
3.1 Data 14
3.2 Automatisk bearbetning 15
3.3 Guldstandard 19

4 Resultat **22**

5 Diskussion och framtida forskning **25**

6 Slutsats **27**

Litteraturförteckning **28**

Förord

Jag skulle i första hand vilja tacka min handledare Beáta Megyesi för hennes engagemang under denna långa resa. Ditt tålamod är inspirerande. Jag vill även visa min tacksamhet till professor Joakim Nivre som har funnits tillgänglig för konsultation och gett sitt stöd när så behövdes. Tack så till min kamrat under projektets första halva, Matilda Bengtsson, för det arbete vi har gjort tillsammans för guldstandarden. Jag hoppas att det var lika lärorikt för oss båda! Slutligen vill jag tacka Maria Kvist och Martin Hassel på DSV i Stockholm som förhoppningsvis ska ha lite nytta av komponenterna som har sammanställts under detta arbete.

1 Inledning

I takt med datoriseringen i samhället har möjligheten att göra patientjournaler tillgängliga elektroniskt för allmänheten också blivit en aktuell fråga. För många människor skulle det vara intressant att på egen hand ta del av sin patientjournal hemma. Det kliniska språket, som återfinns förutom i patientjournaler även i exempelvis dagsanteckningar och läkarrapporter, ställer dock till en del problem gällande begripligheten, då detta språk är främst avsett för kommunikation vårdpersonalen emellan. Kliniskt språk är informationstätt, fragmenterat, men ändå formellt, och undersökningar visar att det finns svårigheter hos patienter med att förstå sina journaler. Detta skapar behov av förenklingssystem, som på automatisk väg skulle kunna ändra och anpassa texten i patientjournaler, för att på så sätt göra dem mer lättlästa och tillgängliga för allmänheten.

Innan ett sådant system för förenkling av språket i patientjournaler faktiskt kan konstrueras, måste mängder av underliggande komponenter falla på plats. Det problematiska i det kliniska språket för en dator kan jämföras med svårigheter som en människa möter vid läsning av samma text: den telegrafiska stilen, mängder av förkortningar, specifika medicinska ord och termer är några exempel. Ett system som är anpassat för arbete med tidningstexter kan följaktligen inte utan vidare vändas om till att förenkla språket i patientjournaler. Likt en människa, behöver ett system i regel anpassas till att arbeta inom en viss språklig domän. Men oavsett domän kan automatisk bearbetning av naturligt språk ses som en kedja av uppgifter som behöver lösas i tur och ordning.

En text som ett system ska arbeta med behöver först genomgå förbearbetning, där till exempel relevanta textstycken skiljs från icke-önskvärd data, som personuppgifter eller datum. Under förbearbetningen löses även andra praktiska problem, såsom inställning av lämpligt filformat för texten att sparas i, och liknande.

Efter förbearbetningen behöver texten tokeniseras och meningssegmenteras, vilket i princip innebär att texten på ord- samt på meningsnivå redigeras till att varje ord hamnar på en egen rad och varje mening avgränsas med en tom dito.

Nästa steg under den automatiska bearbetningen är den morfologiska analysen. I detta steg undersöks texten på ordnivå och märks ut med de enskilda löpordens ordklasstillhörighet och eventuellt även morfologisk information, till exempel tempus på verb eller numerus på substantiv.

Efter att texten har blivit rengjord, tokeniserad och analyserad morfologiskt, är det dags för den syntaktiska analysen, den så kallade parsningen. Vid detta tillfälle analyseras texten strukturellt på

meningsnivå med avseende på syntax enligt en viss lingvistisk teori. Rent grafiskt liknar resultatet för syntaktisk analys av en mening ett slags träd, med bindningarna mellan orden som grenar; därav begreppet syntaxträd.

Innan meningar kan förenklas till att passa en bredare läsargrupp kommer fler steg behövas, men denna uppsats behandlar automatisk bearbetning av språket i patientjournaler till och med den syntaktiska analysen. Det bör nämnas att de ovannämnda länkarna av kedjan, som bygger upp denna uppsats, består dels av fritt tillgängliga verktyg för språkbearbetning, dels av andra som har fått framtas och anpassas för hand.

Ytterligare en aspekt i konstruktionen av ett system för automatisk bearbetning av naturligt språk handlar om systemets utvärdering. För att förbättra parsningsresultaten i synnerhet och ett eventuellt förenklingssystem i allmänhet, behöver man förstås först veta hur bra parsningen har blivit och vilka konkreta problem som systemet har stött på. Utvärderingen av systemet görs genom att jämföra dess slutresultat (i fallet med denna uppsats – syntaxträden av meningar i patientjournaler) med en så kallad guldstandard. En guldstandard är i det här fallet en syntaxträdbank som innehåller samma data som finns i systemets slutresultat, men som har blivit rättad för hand. Ett utvärderingsverktyg kan då jämföra de 2 filerna, visa på konkreta skillnader samt räkna ut ett procentvärde för samstämmigheten mellan systemets resultat och guldstandarden.

Sammanfattningsvis, kommer den här uppsatsen att beskriva en utarbetad kedja av program, med hjälp av vilken rå journaltext, via förbearbetning, tokenisering/meningssegmentering och slutligen morfosyntaktisk analys, omformas till syntaxträd, utvärderade via en korrekt annoterad korpus av kliniska texter, som tas fram för det ändamålet.

1.1 Syfte

Syftet med denna studie är att anpassa existerande verktyg till det kliniska språket i patientjournaler, närmare bestämt röntgenrapporter på svenska, med långsiktigt mål att bana väg för automatisk förenkling av kliniska texter. Arbetet innebär anpassning av tokenisering och meningssegmentering till klinisk text, användning och utvärdering av dagsaktuella ordklasstaggare, samt utveckling av en guldstandard med syntaktiskt annoterad datamängd för att kunna träna och utvärdera parsningsprogram på svensk klinisk text. Alla resultat presenteras på basis av öppen källkod.

1.2 Upplägg

Uppsatsen är indelad enligt följande: i kapitel *2* presenteras tidigare studier kring det kliniska språket och dess bearbetning ur ett språkteknologiskt perspektiv. I kapitel *3* ges beskrivningen av datan som har använts i detta arbete, samt skildringen av den skapade programkedjan, med parsningsresultaten som presenteras i kapitel *4*. Kapitel *5* innehåller vidare diskussion kring de av uppgiften berörda problemen samt en framåtblick, med slutligen kapitel *6* som redogör för de slutsatser som har dragits av arbetet med denna kandidatuppsats.

2 Bakgrund

I den första delen av detta kapitel presenteras tidigare studier kring egenskaperna av det kliniska språket. I den andra delen av kapitlet ges bakgrunden till det kliniska språkets bearbetning ur ett språkteknologiskt perspektiv i form av 4 huvudmoment som denna bearbetning består av inom ramarna för denna studie: förbearbetning, tokenisering/ meningssegmentering, ordklasstaggning, parsning.

2.1 Kliniska texter

Klinisk text produceras av vårdpersonalen för att dokumentera patientens hälsotillstånd och inkluderar exempelvis dagsanteckningar, röntgenrapporter, operationsjournaler, med mera. Dessa texter är skrivna på ett yrkesspråk vars huvudsakliga syfte är anteckningshållning och informationsöverföring vårdpersonalen emellan. Kliniska texter skiljer sig från det generella språket i både struktur och innehåll. Stilen är tämligen fragmenterad, med behov att förmedla komplicerade *yrkesbegrepp* utan långa förklaringar (ex: *Ingen hydronefros. Cirkulation detekteras bilateralt.*). Det kliniska språket som återfinns i patientjournaler är likaså rikt på medicinsk terminologi tack vare kraven på informativitet och precision. Termerna och uttrycken har även ofta utländskt ursprung (ex: *levern försörjs arteriellt via hepatica propria*), framförallt från latin och grekiska (Allvin 2010; Smith 2014; Fan m. fl. 2013; Kvist & Velupillai 2013). Eftersom vedertagna böjningsformer för utländska ord ofta antingen inte är välkända eller helt saknas, återfinns orden inte allt för sällan i ovanliga böjningsformer (ex: *caudalt i vänster njure*; *distal radius ledytan ligger mot benets längsaxel*).

Skrivning i journalerna sker ofta under tidspress, därav är det vanligt förekommande att inmatningshastigheten står över de grammatiska reglerna (Meystre m. fl. 2008; Skeppstedt 2013). Tidspressen och den höga informationstätheten bäddar för användning av *förkortningar*, något som är mycket vanligt i patientjournaler (ex: *pat – patient*; *ua – utan anmärkningar*). Dessa följer ofta inget vedertaget format, utan kan skapas fritt, med fall där samma ord kan förkortas på olika sätt beroende på kontext. Allvin (2010) visar i en journalundersökning att andelen förkortningar varierar mellan 3 och 8 % av det totala antalet ord, samt att förkortningar är även det som oftast väcker osäkerhet i förståelsen från läsarens sida. Anmärkningsvärt är också att förkortningar inte är ett problem enbart för patienter, utan är något som även vårdpersonalen ofta finner problematiskt. Studien av Aantaa (2013) bekräftar att förkortningar som även återfinns utanför den medicinska domänen,

såsom för tid och plats, sällan väcker några svårigheter för lekmän, medan kliniska förkortningar tvingar gissning från kontext vilket leder till misstolkningar, något som gäller fackspråk i allmänhet.

Språket i patientjournaler är inte bara anmärkningsvärt på lexikal nivå, såsom inom förkortningar och terminologi, utan även i syntaktisk struktur. En vanlig egenskap i patientjournaler är till exempel bortfall av *subjekt*. Enligt Haverinen m. fl. (2009) saknades subjekt i 43 % av meningar i en undersökt datamängd. Motsvarande bekräftas av Allvin (2010), som understryker att patienten, som journalens övergripande subjekt, tenderar att vara överflödig att skriva ut under tidspressade förhållanden. I kortare anteckningar med en rubrik kan subjektet dessutom antas vara underförstått från texten (ex: *Även f ö oförändrat.*) eller vara onödigt att skriva ut vid beskrivning av vårdpersonalens egna handlingar (ex: *Inga dislocerade revbenfrakturer ses.*). Undersökning av Aantaa (2013) visar dock att bortfall av subjekt inte skapar svårigheter för läsare kring tolkningen av vem eller vad som är det faktiska subjektet i en sats.

Verb har en intressant ställning i patientjournaler. Bortfall av hjälpverb, vilka sällan har högt informationsvärde, är frekvent, och så är även passiva verbformer (ex: *Tillägg är gjort till svaret.*) (Aantaa 2013; Kvist & Velupillai 2013). Eftersom patienten sällan omnämns som subjekt, blir den passiva formen lämplig. Genom användning av passiva verb kan man även på ett psykologiskt sätt ta avstånd från de beslut som måste fattas vid arbete som läkare, beslut som ibland är svåra (Aantaa 2013). Ofta är det dessutom okänt eller oviktigt vem som utför en viss procedur. Även bortfall av predikat är vanligt i patientjournaler: enligt Haverinen m. fl. (2009) saknades predikat i en tredjedel av meningarna i en undersökt datamängd (ex: *I övriga segment normala fynd.*).

Egenskaperna som har nämnts ovan kan alla finnas sida vid sida inom samma text. Det som således kan härledas är att patientjournaler inte är adresserade till patienten, där de ogrammatiska och telegrafiska språkkonstruktioner, ofta med underförstådd information, försvårar för människor som är intresserade av att själva ta del av sin journal.

2.2 Verktyg

De ovannämnda dragen är inte bara problematiska för mänskliga läsare, utan detta gäller även maskiner. De flesta av dagens verktyg för bearbetning av naturliga språk baseras på tidningstext och är inte välanpassade för arbete med det kliniska språket (Kvist & Velupillai; 2013). På grund av att klinisk text skiljer sig lexikalt och strukturellt från allmänspråket, når verktygen som är tränade på generellt språk inte samma prestanda. Dessa måste utvärderas och adapteras till kliniska texter. De utgör en kedja där det ena bygger på de andras resultat –

det enskilda verktygets korrekthet baseras på de tidigare verktygens korrekthet i kedjan.

Förbearbetning

Det första steget under maskinbearbetning av text på naturligt språk är förbearbetning, även kallad normalisering. Rent översiktligt är målet med förbearbetningen att organisera texten inför vidare processer, och all påföljande analys av texten kommer att bygga på förbearbetningen. Vad termen innebär i praktiken beror till största delen på hur den aktuella texten ser ut, samt på den faktiska planen inför arbetet med texten – det finns således inga definitiva riktlinjer för vad förbearbetning av en text ska innefatta. Det som kan innefattas är avskiljning eller redigering av frekvent förekommande, men icke-önskvärda dataenheter i texten, till exempel personuppgifter och metadata. Det kan också inbegripa normalisering av siffror till ett enhetligt datavärde (ex: <*num*>), borttagning av diakritiska tecken, justering av versaler till gemener eller rättstavningskontroll (Meystre m. fl. 2008). Förbearbetning kan även innebära större textförändringar, exempelvis bortplock av överskrifter i en text, eller handla om rent praktiska frågor som omlagring av texten till ett annat filformat.

Eftersom förbearbetning är en så pass generell, men samtidigt situationsberoende operation, löses den oftast på plats med hjälp av enklare specialskrivna skript. Skriptet skrivs då för att lösa ett eller flera konkreta problem med texten, och förbereda den inför påföljande bearbetning.

Tokenisering och meningssegmentering

Efter att texten har blivit förbearbetad följer dess uppdelning i token – strängar av ett eller flera tecken som tillsammans bildar en enhet. Dessa representeras då inte enbart av enskilda ord, utan även av interpunktion och siffror. Den naturliga avgränsningen mellan token i en text är mellanslag och radbrytning, med mer komplicerade fall som också förekommer. I en tokeniserad text är token i typiska fall avskilda via radbrytning, med varje token på en egen rad. Senare analyser på olika lingvistiska nivåer bygger på korrekt utfört tokenisering.

Meningssegmentering är avgränsning på en högre nivå, meningarna emellan. Svårigheten är att se var en mening slutar och nästa börjar ligger i tvetydigheten hos interpunktionen. Punkt, som den naturliga meningsavgränsaren, är tvetydig och fungerar även som brytmarkör i många förkortningar, efter vilka en mening inte nödvändigtvis behöver ta slut. Ytterligare en svårighet med meningsgränser är avsaknad av interpunktion i meningsslutet, till exempel i en överskrift. Att meningsgränserna har identifierats korrekt är väsentligt för den syntaktiska analysen eftersom syntaxträden byggs på meningsnivå.

Rörande verktyg, liknar situationen med tokenisering och meningssegmentering den med förbearbetning. Kraven kring tokenisering och meningssegmentering kan variera kraftigt beroende på uppdraget i fråga, vilket har gjort att det finns en uppsjö av verktyg, de flesta

framtagna för internt bruk. Ett verktyg vid namn Svannotate* utvecklades vid Institutionen för lingvistik och filologi, Uppsala Universitet, år 2009 som ett paket för bland annat tokenisering och meningssegmentering. Verktygets medföljande dokumentation beskriver de 3 stegen som Svannotate kan utföra, där texten tokeniseras/meningssegmenteras via inbyggda skript, analyseras morfologiskt genom verktyget HunPos och parsas med programmet MaltParser. Således utförs endast tokenisering och meningssegmentering internt av Svannotate, med de 2 påföljande stegen som sker via anrop av externa program. De 3 stegen kan dock utföras separat, ty varje steg lämnar efter sig en utfil med fram till det steget bearbetade texten.

Ordklasstaggning

Nästa steg i bearbetningen behandlar den morfologiska analysen av texten. Orden grupperas traditionellt in i ordklasser, vilka kan variera till antalet. Morfologisk analys inkluderar förutom uppdelning av ord i olika ordklasser även mer eller mindre detaljerad information om morfologiska egenskaper hos ord (exempelvis ordets kasus, numerus eller tempus). Detta är kopplat till begreppet *taggуppsättning* – uppsättning av tillgängliga ordklassorter. Den mer detaljerade morfologiska informationen brukar också ingå i uppsättningen, men antalet tillgängliga taggar kan variera kraftigt mellan olika taggуppsättningar.

Under taggningen går en ordklasstaggare igenom texten och märker ut löporden med morfologisk information utifrån den valda taggуppsättningen. Svårigheten för automatisk ordklasstaggning ligger i att lösa tvetydigheten hos homografer: ord, skrivna på samma sätt, kan höra till olika ordklassorter (ex: *vara, händer, såg*), och ordklasstaggningsprogrammet måste välja ut rätt tagg utifrån den kontext i vilken orden förekommer i.

Programmen baseras på regelbaserade eller statistiska algoritmer (Jurafsky & Martin 2008). Regelbaserade taggare använder databaser med stora mängder handskrivna regler som beskriver exempelvis när ett visst tvetydigt ord hör till en viss ordklass när det följs eller föregås av ett annat ord med en viss ordklass. På senare tid har statistiska taggare tagit täten. Dessa är datadrivna, det vill säga de använder en träningskorpus där orden i meningar har blivit taggade för hand i förväg, alternativt automatiskt med påföljande validering – eller en kombination av dessa. En statistisk taggare skapar en modell av träningskorpusen och får på det sättet sannolikheten för att ett visst ord i korpusen ska ha vissa morfologiska egenskaper. Exempel på en träningskorpus för svenska är Stockholm-Umeå Corpus (SUC), som består av cirka en miljon löpord, med texter i olika genrer. Taggуppsättningen i version 3.0 av SUC består av 153 ordklasstaggar (Östling; 2013).

För allmän svensk text ligger i dagsläget de bästa ordklasstaggningsresultaten på kring 97 %. Exempel på statistiska

* Filip Salomonsson, Institutionen för lingvistik och filologi, Uppsala Universitet

ordklasstaggare är verktyg som Stagger[1] och HunPos[2]. Stagger är utvecklad för svenska och visar en prestanda på 96,6 % per löpord (Östling; 2013) med modell tränad på SUC 3.0. Även ordklasstaggaren HunPos, trots att dess utveckling har på senare tid stannat av, visar på motsvarande siffror (Halacsy m. fl. 2007; Megyesi 2007).

Dessa siffror gäller dock inte för taggning av kliniska texter: studien av Coden m. fl. (2005) visar en prestanda på 87 % när en allmän engelskspråkig korpus användes. Motsvarande siffror nåddes för svenska med SUC 3.0 som referens (Smith; 2014). Ferraro m. fl. (2013) kom i en liknande studie fram till nivån av 88,6 %. Högre siffror kunde nås med inlägg av domänspecifika kliniska korpusar, vilket tyder på problem med det medicinska ordförrådet som grund. Ändå tycks inte klinisk text skilja sig allt för mycket morfologiskt från generell svenska, vilket gör att existerande ordklasstaggare lämpar sig för ordklassanalys (Hassel m. fl. 2011).

Parsning

Slutsteget i morfosyntaktisk analys är den syntaktiska analysen, också kallad parsning. Den används för att fastställa relationer mellan orden i en mening utifrån en given grammatik.

Man kan skilja mellan 2 populära grammatiska teorier som används för syntaktisk analys inom bearbetning av naturligt språk: frasstrukturgrammatik och dependensgrammatik. Båda grammatiker avbildar meningar genom syntaxträd, men inom dependensgrammatiken utgörs trädens alla noder av meningens ord, medan frasstrukturgrammatiken ser icke-terminala noder som fraser (exempelvis som verb- eller nominalfraser). Frasstrukturgrammatiken, där meningen alltså ses bestå av fastare enheter, har svårt att hantera språk med friare ordföljd. Därav blir det mer fördelaktigt att avbilda meningar med hjälp av dependensgrammatik. Detta gäller även för kliniska texter med deras fragmenterade struktur, följaktligen även för språk med annars relativt fast ordföljd (Meystre m. fl. 2008).

Parsning kan jämföras med en slags sökning. Detta lämnar plats för 2 basalgoritmer: toppen-nedåtsökning och botten-uppåtsökning. Vid toppen-nedåtsökningen försöker parsern bygga delträd utifrån grammatikreglerna med start i toppnoden, och nedåt till löven, tills ett träd som matchar hela den faktiska meningen är byggd. Vid botten-uppåtsökningen bygger parsern träd från vänstra nedre lövet och arbetar uppåt mot roten. Båda sökmetoder för att bygga alla möjliga syntaxträd för en mening har sina för- och nackdelar. Toppen-nedåt kommer inte att bygga träd som inte leder till toppnoden, men däremot en del träd som inte är konsistenta med orden i meningen. Botten-uppåt kan endast producera konsistenta träd eftersom sökningen börjar i löven, men dock kan träd som inte leder till roten skapas (Jurafsky & Martin 2008).

Parsningsprogram kan i likhet med ordklasstaggare vara regelbaserade eller statistiska, där de senare är de numera mest

[1] ling.su.se/english/nlp/tools/stagger/stagger-the-stockholm-tagger 04.09.14

[2] code.google.com/p/hunpos 04.09.14

använda. De statistiska parsrarna förutsätter därav en språkmodell som har blivit tränad på en annoterad text, med alla relationer mellan orden utsatta i förväg. Till aktuella verktyg för den syntaktiska analysen hör MaltParser[1] och MSTParser[2]. Båda är språkoberoende och datadrivna, vilket vid tillgång till tillräckliga datamängder möjliggör användning inom valfria språk/subspråk givet en syntaktiskt annoterad datamängd som parsern kan tränas på (Nivre m. fl. 2007). MaltParser har testats på en mängd språk, med en prestanda på i snitt 80 % (Nivre; 2008). MSTParser tillämpar grafbaserad ansats till dependensparsning genom exakta inferensalgoritmer. Likt MaltParser, har MSTParser öppen källkod (McDonald m. fl. 2005). Vid konferensen CONLL 2006 har de 2 parsrarna visat bäst resultat och har därefter använts för ett stort antal naturliga språk (Buchholz m. fl. 2006).

Begränsad tillgång till syntaktiskt annoterade kliniska texter, ihop med de facto frånvaro av standarder för sådan annotering, sätter dock käppar i hjulet för utvecklingen av parsrar för den medicinska domänen, samt försvårar resultatjämförelser mellan system. Som motsats, har de generella parsrarna tillgång till större datasamlingar med syntaktiskt annoterat material, fulländade med klara riktlinjer för vidare annotering (Fan m. fl. 2013).

Till vanliga fel inom syntaktisk parsning av kliniska texter hör missar rörande konjunktioner, adverbial och prepositioner. Skeppstedt (2013) uppmärksammar ett antal feltyper: förkortningar med punkt kan misstolkas för att enbart vara substantiv; felaktig etikettering av adjektiv (ofta på grund av förkortning av de senare); svårigheter med adverbialsorter (såsom tids- eller platsadverbial); feltolkning av första ordet i ett flerordsuttryck som determinerare eller objekt som meningens subjekt. Meningslängd har även visat sig vara en försämrande faktor rörande parsningskvaliteten, med kortare meningar oftare analyserade felfritt. För ofullständiga meningar saknas det idag en allmänt vedertagen lösning kring frågan om syntaktisk representation – till exempel om avsaknad av subjekt eller predikat ska märkas ut med en attrapp eller ej. Frågan gäller även okända förkortningar, samt till exempel huruvida uttryck på andra språk än textens huvudspråk ska märkas ut med en enhetlig tagg eller om de bör översättas (Fan m. fl. 2013).

Nästa avsnitt behandlar den använda kliniska textdatan, tillsammans med beskrivningen av verktygen som har anpassats till klinisk text.

1 maltparser.org 04.09.14

2 seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html 04.09.14

3 Metod

I denna studie anpassas existerande språkbearbetningsverktyg, utvecklade för att analysera allmänsvenska, till det kliniska språket i patientjournaler. Arbetet innefattar anpassning av tokenisering och meningssegmentering till klinisk text, adaptering och utvärdering av ordklasstaggare, samt utveckling av en guldstandard med syntaktiskt annoterad datamängd för träning och utvärdering av parsningsprogram.

I kapitlets första del beskrivs den kliniska textdatan som har använts inom studien. I den andra delen av kapitlet presenteras den utarbetade programkedjan, med hjälp av vilken rå journaltext, via förbearbetning, tokenisering/meningssegmentering och morfosyntaktisk analys, kan annoteras med syntaktisk information. Kapitlets tredje del beskriver guldstandarden som har tagits fram för att kunna träna och utvärdera parsrar på svensk klinisk text.

3.1 Data

Detta arbete bygger på kliniska röntgentexter ur den stockholmska EPR-korpusen* bestående av en miljon elektroniska patientjournaler från det sena 2000-talet. De kliniska röntgentexterna i korpusen består av strax under 450 000 röntgenrapporter från åren 2009-2010, och inbegriper ungefär 150 000 patienter. Texterna utgörs av cirka 10 miljoner löpord utspridda över 1,1 miljon meningar. Datasamlingen i det aktuella arbetet är uppdelad i medicinska bedömningar – avsnitt av varierande längd från någon enstaka fras till stycken på tiotals ord. Språket är typiskt för att vara kliniskt, med fragmenterade, tämligen korta och ofta verblösa satser, kombinerade med frekventa förkortningar. Exempel 1 återger hur data ser ut med viss modifiering och efter anonymisering:

Ingen skelettskada.

Ingen blödning. Ingen infarkt påvisad. Normalvida likvorrum. Ingen expansivitet.

Levertransplantatet har ordinär ekogenicitet utan parenkymförändringar. Det finns inga vidgade intra- eller extrahepatiska gallgångar. Samtliga leverkärl är öppetstående i såväl leverhilus som intrahepatiskt.

Exempel 1. Tre medicinska bedömningar av varierande längd.

* dsv.su.se/en/research/research-areas/health/stockholm-epr-corpus 04.09.14

3.2 Automatisk bearbetning

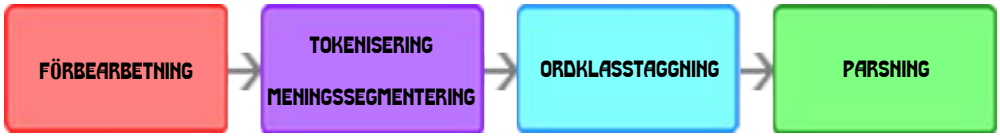
För att kunna ta fram syntaktiskt analyserade meningar av kliniska texter måste verktyg som baseras på allmänsvenska utvärderas och anpassas.

Förbearbetning. Det första steget i programkedjan har som mål att anpassa och förbereda data för vidare analys. Detta tas omhand av ett specialskrivet skript som tar hand om meningsextraktion, tillsammans med datanormalisering som separering av överskrifter och borttagning av personuppgifter.

Tokenisering och meningssegmentering. Syftet med dessa är att förbereda texten inför ordklasstaggning och parsning: felaktig ordgräns får till exempel konsekvensen under den morfologiska analysen i form av felaktig taggning, medan fel meningsgräns under den syntaktiska analysen kan få parsern att slå ihop eller bryta upp meningar som följd.

Ordklasstaggning. Målet med ordklasstaggningen är att märka ut ordens ordklasstillhörighet med eventuella morfologiska egenskaper i en viss kontext i syfte att underlätta för parsern under den syntaktiska analysen.

Slutsteget i programkedjan är *parsning*. Under detta steg använder sig parsern av den ordklasstaggade textfilen i syfte att producera syntaxträd utifrån den givna språkmodellen. Syntaxträden kan sedan exempelvis användas under projekt med mål att på automatisk väg förenkla de kliniska texterna.



Förbearbetning

Kliniska texter som användes var sparade i en tabellfil med flera kolumner, där en av kolumnerna innehöll de relevanta meningarna. Ett extraktionsskript följde med datan, men var främst avsett för att ta ut meningarna ur denna kolumn och skriva ut dem till en ny textfil. Vidare normalisering gjordes med hjälp av ett utökat skript. Modulen tar hand om 3 saker: separering av överskrifter från övrig text, borttagning av personnamn samt borttagning av personnummer. Nedan visas hur exempeltext ser ut innan och efter normalisering:

```
DT Thorax och övre buk
Inga mediastinala expansiviteter.
Enstaka icke-patologiskt förstorade lymfkörtlar i
mediastinum.
Inga lunginfiltrat.
```

```
Patientdata: 123456-7890 Efternamn, Namn
Läkare Namn Efternamn
```

Exempel 2. Text innan förbearbetning.

```
DT Thorax och övre buk

Inga mediastinala expansiviteter.
Enstaka icke-patologiskt förstorade lymfkörtlar i
mediastinum.
Inga lunginfiltrat.
```

Exempel 3. Text efter förbearbetning.

Överskrifter. Dessa är kortare satser på ett eller ett par ord som beskriver huvudstycket. Svårigheten ligger i att avskilja överskrifter från övrig löpande text. Överskrifter är ofta inte separerade från textstycket med en extra radbrytning, utan textstycket börjar i regel på raden direkt efter överskriften. Samtidigt avslutas inte överskrifter med interpunktion. Dessa faktorer gör det omöjligt att exempelvis gå efter radbrytningar för att avskilja överskriften eftersom felaktiga meningsgränser kan då uppstå mitt i en mening ifall skribenten själv började på en ny rad mitt i meningen.

Utgångspunkten för att lösa detta är att använda heuristik: en överskrift består inte av färre än X och fler än Y tecken (de valda längderna nedan är 3-45 bokstäver), dessutom avslutas den inte med punkt, utan med ett nyradstecken. Överskrifter kan bestå helt av versaler (*LUNGOR*), eller av en (*Axel höger*) alternativt flera (*DT Buköversikt*) versaler följda av gemener. Det inledande uttrycket nedan täcker det första och det tredje fallet, medan det andra täcker fall 2:

```
"^([A-ZÅÄÖ]{3,45}.{,45}\n)"
 "^([A-ZÅÄÖ]{1}.{,45}\n)"
```

Personnamn. En annan uppgift är borttagning av personuppgifter. De kliniska texterna innehåller ett stort antal namn, telefon- och även personnummer. För att ta bort namn kan man utgå ifrån tesen om att dessa är minst 2 stycken (för att även täcka mellannamn) påföljande ord som börjar med versaler, samt kan vara separerade med komma: *Förnamn Efternamn* | *Efternamn, Förnamn*. Dessa uppgifter var i regel placerade i ett speciellt stycke efter den medicinska bedömningen, så att en rad som innehåller ett namn kan således tas bort i sin helhet:

```
"\n.{,30}[A-ZÅÄÖ]{1}[a-zéåäö]{1,30}[, ]{1,2}[A-ZÅÄÖ]
          {1}[a-zéåäö]{1,30}.{,30}(\n|$)"
```

Det reguljära uttrycket tillåter upp till 30 tecken efter radbrytning, kräver en stor bokstav samt mellan 1 och 30 påföljande gemener, förutsätter ett mellanslag med eller utan komma, samt ännu en stor

bokstav med mellan 1 och 30 påföljande gemener. Sedan tillåts ytterligare upp till 30 tecken med efterföljande radslut. Tillåtelse av upp till 30 andra tecken före och efter beror på att namnen i personuppgiftsstycket inte alltid står på en egen rad, utan kan blandas med andra personuppgifter som också kunde avlägsnas.

Personnummer. Normaliseringens slutliga steg handlar om personnumrens identifiering. Deras relativt fasta struktur underlättar i detta. Det reguljära uttrycket nedan tillåter 8-10 siffror, med ett möjligt bindestreck före de 4 sista:

```
"([0-9]{2})?[0-9]{6}-?[0-9]{4}"
```

Förbearbetningssteget producerar slutligen en textfil med ur datasamlingen extraherade medicinska bedömningar, som har blivit normaliserade med separerade överskrifter samt borttagna personuppgifter.

Tokenisering och meningssegmentering

Förbearbetningen lämnar efter sig en städad, men annars icke-analyserad text. På grund av förbearbetningens situationsberoende natur, löstes den med hjälp av specialskrivna skript. Därav tillkom behovet av att välja mellan färdiga verktyg från och med programkedjans andra steg, vid tokenisering/meningssegmentering.

Vid Institutionen för lingvistik och filologi har den för svenska utvecklade statistiska ordklasstaggaren Stagger tidigare använts med goda resultat (Smith; 2014). Anledningen till att denne ordklasstaggare blev aktuell redan under det andra steget, är för att Stagger förutom ordklasstaggning även innehåller mekanismer för tokenisering och meningssegmentering.

Samtidigt fanns det även tillgång till det vid samma institution egenutvecklade språkbearbetningspaketet Svannotate. Detta paket innehåller, förutom anrop till ordklasstaggnings- och parsningsverktyg, även egna moduler för tokenisering och meningssegmentering.

Användning av Stagger har dock visat sig vara olämplig på grund av fel i meningssegmenteringen: verktyget visade sig identifiera meningsgränser felaktigt, till exempel i förkortningsavslutande punkter, samt i datum. Detta kombinerades med svårigheter av kontroll över dessa inbyggda mekanismer. Verktyget Svannotate har därav använts istället. Programmet är utvecklat för internt institutionsbruk och saknar utförlig dokumentation, något som kan bli problematiskt vid behov av detaljstyrning. För uppgiften i fråga har dock verktyget visat sig fungera bra, även fast inga anpassningar till arbete med klinisk text hade gjorts. Den normaliserade textfilen från förbearbetningssteget läses in av Svannotate som utför tokenisering/meningssegmentering, och skriver ut resultatet till en ny textfil.

```
DT Thorax och övre buk

Inga mediastinala expansiviteter. Enstaka icke-
patologiskt förstorade lymfkörtlar i mediastinum.
```

Exempel 4. Förbearbetad indata till Svannotate.

```
1  DT
2  Thorax
3  och
4  övre
5  buk

1  Inga
2  mediastinala
3  expansiviteter
4  .

1  Enstaka
2  icke
3  patologiskt
4  förstorade
5  lymfkörtlar
6  i
7  mediastinum
8  .
```

Exempel 5. Tokeniserad och meningssegmenterad utdata från Svannotate.

Ordklasstaggning

Till följd av svårigheter med tokenisering/meningssegmentering i Stagger har användningen av detta verktyg inte visat sig vara lämplig. Anledningen till att användning av Stagger även föll bort under det påföljande steget med ordklasstaggningen, är för att Stagger visade sig utföra båda stegen i ett svep. Eftersom alla steg i kedjan bygger på de föregående stegen, skulle felen i tokenisering/meningssegmentering föras vidare till ordklasstaggningen, med felaktig ord- och meningsgräns som följd. Felen i ordklasstaggningen skulle på sikt även påverka parsningen.

Som alternativ till Stagger utfördes ordklasstaggningen med verktyget HunPos som också i tidigare studier har visat bra resultat (Halacsy m. fl. 2007; Megyesi 2007). Ordklasstaggaren HunPos med tillhörande modell ingår i färdig form i Svannotate och är tränad på Stockholm-Umeå Corpus, version 2.0. Dokumentationen nämner att träningen har gjorts med standardparametrarna i HunPos (*-t 2, -e 2, -f 10, -s 10*). Verktyget Svannotate kan automatiskt anropa HunPos för steget med ordklasstaggningen och direkt förmedla den tokeniserade och meningssegmenterade textfilen.

Under ordklasstaggningssteget används således den tokeniserade och meningssegmenterade textfilen där tokenen via programmet HunPos märks ut med ordklasstaggar. Ytterligare en mindre aspekt är det för nästa steg efterfrågade filformatet: vid krav på CONLL-format för steget med parsningen kan ett konverteringsskript som säkerställer detta användas.

Parsning

För att säkerställa en viss variation av parsningsresultaten har de 2 aktuella språkoberoende verktygen MaltParser och MSTParser använts. Dessa parsrar har visat på god prestanda i tidigare tester (Buchholz m. fl. 2006) för en stor mängd naturliga språk, därav förhoppningen om att användning av MaltParser och MSTParser för analys av det kliniska språket också kan ge tillfredsställande resultat.

För MaltParser användes den förinställda färdigoptimerade modellen *swemalt-1.7.2.mco*[1], tränad på Talbankdelen av den svenska Trädbanken[2]. För MSTParser tränades däremot modellen speciellt för uppgiften, dock på samma träningsmängd som *swemalt-1.7.2.mco* för MaltParser för att säkerställa likartade förhållanden parsrarna emellan. MSTParser har endast 2 parametrar att optimera (*modell*, ordning 1/2 samt *avkodning*, projektiv/icke-projektiv), vilket gör det lämpligt att testa alla 4 kombinationer genom att skapa 4 modeller.

3.3 Guldstandard

Ett system för automatisk bearbetning av naturligt språk behöver kunna utvärderas. Detta kan med fördel göras genom att jämföra systemets parsningsresultat med motsvarande data som har blivit rättad för hand. En guldstandard är således en rättad korpus som anses vara korrekt, och som innehåller samma data som finns i systemets slutresultat, dock manuellt granskad. För framtagning av en sådan guldstandard inom ramarna för detta arbete, rättades ett block på cirka 400 syntaxträd av varierande storlek som hade producerats från meningarna av programkedjans sista länk MaltParser.

MaltParser lagrar de skapade syntaxträden i en textfil. Uppställning i filen är av CONLL-modellen[3] – ett kolumnformat för lagring av meningskonstruktioner:

I	II	III	IV	V	VI	VII	VIII
1	Någon	_	DT	DT	UTR|SIN|IND	3	DT
2	kvarvarande	_	JJ	JJ	POS|NOM	3	AT
3	pneumothoraxspalt	_	NN	NN	UTR|SIN|IND|NOM	4	SS
4	kan	_	VB	VB	PRS|AKT	0	ROOT
5	ej	_	AB	AB	_	4	NA

[1] maltparser.org/mco/swedish_parser/swemalt.html 04.09.14
[2] stp.lingfil.uu.se/~nivre/swedish_treebank 04.09.14
[3] nextens.uvt.nl/depparse-wiki/DataFormat 04.09.14

6	påvisas	_	VB	VB		INF|SFO	4	VG
7	.	_	MAD	MAD		_	4	IP

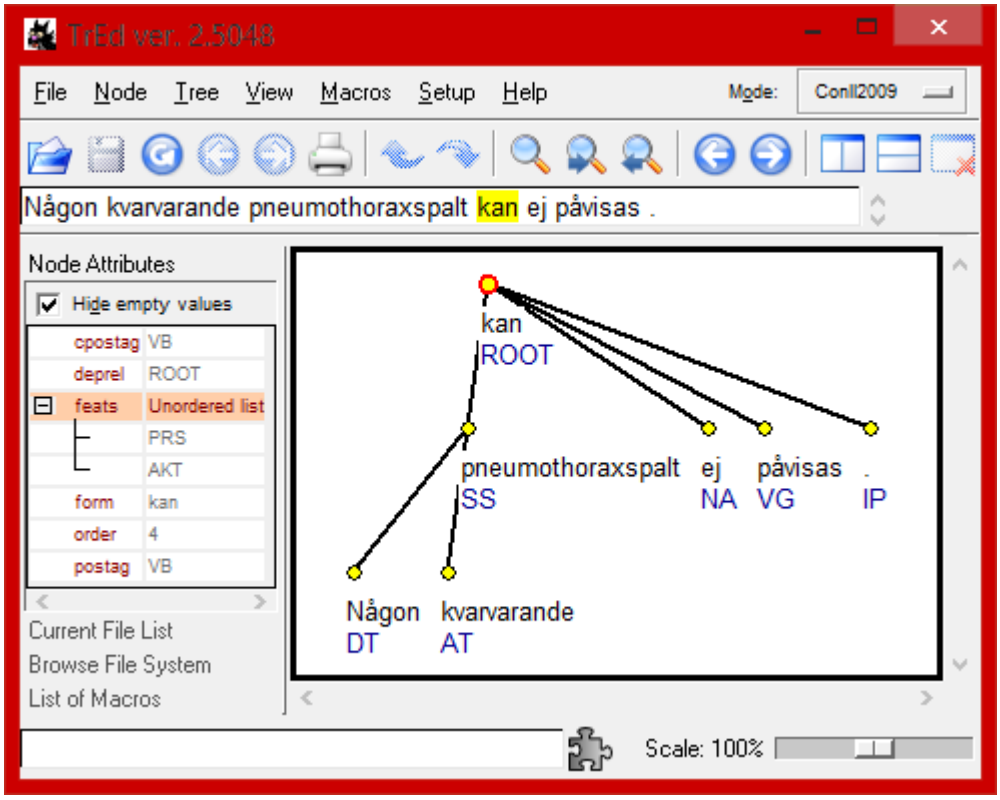
Tabell 1. Exempelmening i CONLL-format.

I tokenets ID
II tokenet som det förekommer i meningen
III lemmaformen av tokenet; understreck vid avsaknad
IV taggуppsättningens övergripande ordklasstagg
V taggуppsättningens fullständiga ordklasstagg; taggen från föregående kolumn vid avsaknad
VI morfologiska egenskaper av tokenet; understreck vid avsaknad
VII dependensrelation av tokenet i form av ID från dess huvudtoken
VIII dependensrelationstagg av tokenet, dess syntaktiska roll i meningen

De 2 sista kolumnerna är de som ändrades vid behov under rättningen. Kolumn 7 visar parsningen i textform – för ordet *någon* innebär siffra 3 kopplingen till det tredje ordet i meningen, *pneumothoraxspalt*. Kolumn 8 med dependensrelationstaggen *DT* visar att ordet i fallet är en determinerare.

För den manuella granskningen, som gjordes tillsammans med en annan student i datorlingvistik, användes programmet TrEd* (Pajas & Fabian 2011), där meningarnas väsentliga *dependensrelationer* (bindningar mellan orden) samt deras *dependensrelationstaggar* (etiketter under orden) granskades. Meningen ovan representeras i Figur 1 i form av ett grafiskt syntaxträd i programmet TrEd.

* ufal.mff.cuni.cz/tred 04.09.14



Figur 1. Bildrepresentation av syntaxträd i TrEd.

TrEd används här endast som ett visualiseringsverktyg för utfiler från parsrar, vilket medför att när ett textsegment, som anses vid rättningen vara fel, upphittas, behöver det letas upp i parsningsfilen för att kunna korrigeras. Guldstandarden är således en CONLL-fil bestående av cirka 400 rättade syntaxträd. Under utvärderingen används guldstandardfilen som jämförelse i förhållande till programkedjans resultat med exakt samma, dock orättade, meningar.

För att slutligen automatiskt sammanställa parsningsresultaten med guldstandarden kan utvärderingsverktyget MaltEval* användas. MaltEval jämför parsrarnas utfiler med guldstandardfilen och räknar ut procentvärdena LAS/UAS för samstämmigheten mellan systemets resultat och guldstandarden. *L*abeled *A*ttachment *S*core är andelen token där både dependensrelationen och dependensrelationstaggen har blivit korrekt analyserade av parsern. *U*nlabeled *A*ttachment *S*core är andelen token där enbart dependensrelationen är kontrollerad till att stämma med guldstandarden. MaltEval producerar även statistisk information kring dependensrelationstaggar i syntaxträden.

Nästa avsnitt behandlar parsningsresultaten från MaltParser och MSTParser, med tillhörande tabeller över dependensrelationstaggarna.

* maltparser.org/malteval.html 04.09.14

4 Resultat

Kapitlet presenterar resultaten från programkedjans slutdel – parsningen. Inom kedjans sista länk har de 2 parsrarna, MaltParser och MSTParser, jämförts. För utvärderingen används en framtagen guldstandard bestående av cirka 400 meningar av varierande meningslängd. Samstämmigheten mellan guldstandarden och parsrarna, samt dependensrelationstaggarna visas i respektive tabeller.

Utvärderingen i arbetet har gjorts med programmet MaltEval. Verktyget är känsligt för skillnader i data, därav hade exakt de meningar som finns rättade i guldfilen klippts ut ur datamängden och körts genom kedjan. MaltEval är byggd för analys av utfiler från MaltParser, men har även använts för MSTParser efter viss anpassning.

För MaltParser användes dess färdigoptimerade modell. För MSTParser nyskapades 4 modeller utifrån parserns 2 optimerbara parametrar (*modell*, ordning 1/2 samt *avkodning*, projektiv/icke-projektiv). Resultat med Labeled/Unlabeled Attachment Score (LAS respektive UAS) för körningarna återfinns i tabellen nedan:

Modell	*LAS*	*UAS*
Malt	**68,8**	**75,5**
MST 1 proj	61,1	71,8
MST 2 proj	**61,6**	**72,3**
MST 1 ej proj	60,9	71,6
MST 2 ej proj	61,2	71,7

Tabell 2. MaltEvals beräkning av LAS/UAS; de högsta värdena är fetstilta.

Resultatet för MaltParser ligger på 68,8 % för LAS och 75,5 % för UAS. MSTParser får resultat på mellan 60,9 och 61,6 % för LAS samt på mellan 71,6 och 72,3 % för UAS, där modellen MST 2 proj visar de högsta värdena. Märkbart är även att skillnaden hos MaltParser mellan LAS och UAS är tydligt mindre än motsvarande för MSTParser (6,7 % kontra ≥10,5 %).

Dependensrelationstaggarna, som också fick en genomgång under framtagningen av guldstandarden, presenteras nedan. Tabellen visar uppställningen av taggarna från körningen med MaltParser, samt MSTParser med de högsta LAS-/UAS-värden (projordning 2). Tabellen består av 5 kolumner:

- *deprel*: beteckning på dependensrelationstaggen[⋆]
- *precision*: antal taggar som hade blivit kategoriserade rätt, dividerat med antalet taggar som faktiskt kategoriserades
- *täckning*: antal taggar som hade blivit kategoriserade rätt, dividerat med det totala antalet taggar

⋆ stp.lingfil.uu.se/~nivre/swedish_treebank/dep 04.09.14

- *f-värde*: harmoniskt medelvärde av precision och täckning
- *antal*: mängd förekomster inom datamängden över de av parsrarna identifierade taggarna

deprel	*precision*		*täckning*		*f-värde*		*antal*	
	MALT	MST	MALT	MST	MALT	MST	MALT	MST
ROOT	0,778	0,752	0,780	0,752	0,779	0,752		388
+A	0,923	0,611	0,667	0,611	0,774	0,611	13	18
AA	0,770	0,589	0,676	0,495	0,720	0,538	166	160
AG	1,00	0,75	1		1,000	0,857	6	8
AN	0,444	0,118	0,857	0,143	0,585	0,129	100	17
AT	0,858	0,834	0,864	0,843	0,861	0,839	289	290
CA	0,789	0,500	0,833	0,611	0,811	0,550	19	22
CJ	0,684	0,492	0,646	0,578	0,665	0,531	152	189
DT	0,790	0,796	0,935	0,848	0,856	0,821	749	719
ES	0,750	0,625	1,000	0,556	0,857	0,588	12	8
ET	0,777	0,697	0,886	0,741	0,828	0,718	251	234
+F	0,200	0,167	1,0	0,5	0,333	0,250	30	18
FS	1		0,833	0,750	0,909	0,857	10	9
HD	0,700	0,537	0,497	0,385	0,581	0,448	120	121
I?	1		1		1			5
IF	0,7		0,875		0,778			10
IK	1,000	0,945	0,873	0,945	0,932	0,945	48	55
IP	0,997	0,994	1		0,998	0,997	310	311
IQ	1		1		1			45
IR	1		1		1			9
IT	1		0,714		0,833			10
IU	1		1		1			1
JR	1		0,778	1,000	0,875	1,000	7	9
KA	0,750	0,364	0,857	0,571	0,800	0,444	8	11
MA	0,818	0,714	0,750	0,417	0,783	0,526	11	7
MS	0,535	0,32	0,767	0,533	0,63	0,40	44	52
NA	0,857	0,812	0,60	0,65	0,706	0,722	14	16
OA	0,880	0,646	0,917	0,646	0,898	0,646	50	48
OO	0,694	0,554	0,608	0,474	0,648	0,511	85	83
OP	0,750	0,333	1,000	0,333	0,857	0,333	4	3
PA	0,819	0,738	0,918	0,903	0,866	0,812	370	404
PL	0,846	0,750	1,000	0,818	0,917	0,783	35	34
PT	0,625	0,357	0,385		0,476	0,370	8	14
RA	0,885	0,632	0,670	0,466	0,762	0,536	78	76
SP	0,814	0,521	0,921	0,658	0,864	0,581	43	48
SS	0,791	0,699	0,716	0,637	0,751	0,667	172	173
TA	0,585	0,479	0,333	0,319	0,425	0,383	41	48
UA	0,618	0,522	0,778	0,444	0,689	0,480	34	23
VA	0,8	0,6	0,667	0,500	0,727	0,545		5
VG	0,939	0,833	0,705	0,682	0,805	0,750	33	36

Tabell 3. Resultaten för dependensrelationstaggar.

Tabellen visar att de 4 dependensrelationstaggarna IQ, IR, IU, I? (alla beskrivande interpunktion) har 100 % samstämmighet med guldstandarden för båda parsrar. Ytterligare interpunktionstaggar (IK, IP) har nära hundraprocentig samstämmighet. Värden från kring 80 % och uppåt över både precision, täckning och f-värde visar även agent (AG), adjektivbestämning (AT), determinerare (DT), attrappsubjekt (FS) samt prepositionsfyllnad (PA).

Taggarna med motsvarande antal förekomster, men tydlig skillnad i värden mellan de 2 parsrarna kan uppmärksammas: MaltParser klarar bättre av kategorierna kontrastivt adverbial (CA; f-värde – Malt: 81 % | MST: 55 %), logiskt subjekt (ES; 86% | 59%), komparativt adverbial (KA; 80% | 44%), objektspredikativ (OP; 86% | 33%), subjektivt predikativkomplement (SP; 86% | 58%). Eftersom MSTParser har överlag sämre värden än MaltParser finns det få exempel där taggning gjort av MSTParser har högre samstämmighet med guldstandarden än MaltParser. Taggen som främst kan nämnas hör till interpunktioner – högra parentesen (JR; 87,5% | 100%). En liten skillnad, under 2 % i f-värde till MST-fördel, visar även taggen negationsadverbial (NA; 70,6% | 72,2%).

Bland de svårare taggarna för parsrarna, med siffror överlag under 50 % på precision, täckning och f-värde, kan ses huvudsatssamordning (+F; 33% | 25%), predikativattribut (PT; 48% | 37%) samt tidsadverbial (TA; 42% | 38%). Extralåga siffror för MSTParser jämfört med MaltParser visar taggarna apposition (AN; 58% | 13%), samordnad konjunktion (CJ; 66% | 53%), makrosyntagm (MS; 63% | 40%). För AN bör tilläggas den sexfaldiga skillnaden mellan parsrarna över antalet ord som identifierades som denna tagg, 100 för Malt mot 17 för MST.

Det kan konstateras att MaltParser ger i överlag bättre resultat i jämförelse med MSTParser när den tränas och testas på samma datamängder.

Nästa avsnitt innehåller en diskussion kring parsningsresultaten, samt berör möjlig framtida forskning.

5 Diskussion och framtida forskning

I detta kapitel förs diskussion rörande programkedjans resultat, därav i jämförelse med allmänsvenska, tillsammans med ideer kring framtida forskning.

Inom ramarna för denna studie har det för parsningen med MaltParser använts en existerande modell, tränad på Talbanken. För MSTParser fanns inte en existerande modell att tillgå, därav tränades nya modeller och parserns alla 4 parameterkombinationer testades. Skillnaderna mellan parsrarnas resultat ligger trots det tydligt till MaltParsers fördel, med strax över 7 % i LAS jämfört med MSTParsers bästa modell (3,2 % till MaltParsers fördel i UAS).

De generella LAS-resultaten på 68,8 % för MaltParser och 61,6 % för MSTParser borde även ses i jämförelse med allmänsvenska. I tidigare studier har MaltParser och MSTParser visat värden på 84,5 % respektive 82,5 % för allmänsvenska (Nivre & McDonald; 2008) – resultat som således är över 15 % högre än dito för kliniska röntgentexternas analys i denna studie:

	röntgentexter	*allmänsvenska*
MaltParser	68,8	84,5
MSTParser	61,6	82,5

Tabell 4. LAS-värden för parsning av kliniskt respektive allmänt språk.

Orsaken kan dels vara det fragmenterade kliniska språket i sig. Jämförelse av dess analys med analys av allmänsvenska kunde förväntas vara till det senares fördel, eftersom språkbearbetningsverktyg är överlag byggda för att främst hantera det allmänna språket. Det ska dock tilläggas att dessa värden är inte helt jämförbara, ty modellerna som ligger till grund för siffrorna för röntgentexter respektive allmänsvenska är ej tränade på samma datamängder.

En ytterligare orsak kan finnas i guldstandarden. Det kan tänkas att tillgång till djupare lingvistiska kunskaper under dess utveckling hade varit till fördel, även om det är svårt att säga hur mycket eventuella annoteringsfel i den framtagna guldstandarden kunde försämra parsningsresultaten. En mer genomarbetad guldstandard är dock ett exempel på lämplig vidare utveckling.

Det som också skulle kunna undersökas vidare är om det kan finnas problem med den morfologiska analysen. Det kan vara lämpligt att titta närmare på möjligheten att implementera Stagger inom ramarna för programkedjan och testa om detta kan gynna resultaten av parsningen. Eftersom problemen med Stagger handlade om svårstyrd felaktig tokenisering/meningssegmentering, kan lösningen tänkas ligga i förbearbetningen som på ett systematiskt sätt skulle kunna anpassas för Stagger och på så vis minska risken för fel.

Tidigare studier kring dependensrelationer visar även generellt att de typer (såsom interpunktion IQ, IR), vars tokenhuvud tillhör de stängda ordklasserna, oftare analyseras korrekt (Nivre m. fl. 2007). Detta gäller även till exempel adjektivbestämningen (AT). Till de svårare klasserna hör exempelvis infinitiva komplement och appositioner, där de senare, ihop med objektiva predikativ (OP), tenderar att vara extrasvåra för MSTParser. Dessa resultat kunde även bekräftas i denna studie, då motsvarande dependensrelationstaggar visade likartade mönster inom de använda parsrarna.

Framtida arbete, till exempel rörande förbättring och utökning av guldstandarden, kan inbegripa minimalt övervakad inlärning, så kallad *bootstrapping*. Med detta skulle utökningen av guldstandarden kunna automatiseras genom att med en mindre mängd annoterad data tagga ny data och träna om. Även träning av nya parsningsmodeller med genomarbetad träningsdata, som är mer anpassad till den givna språkdomänen, kan vara lämpligt att undersöka.

Nästa kapitel innehåller slutsatsen av detta arbete.

6 Slutsats

Möjligheten att kunna ta del av sin patientjournal elektroniskt på egen hand har idag blivit ett aktuellt ämne. Det fragmenterade kliniska språket i patientjournalerna har dock i tidigare studier visat sig vara ett hinder för många patienter, med svårigheter att förstå informationen i journalerna som följd. Detta leder till behov av system för automatisk förenkling av det kliniska språket.

I denna kandidatuppsats har en kedja av verktyg och skript för förbearbetning, tokenisering, meningssegmentering, ordklasstaggning och parsning av kliniska texter sammanställts, samt översiktligt utvärderats mot en mindre guldstandard. Utvärderingen har inbegripit jämförelse av 2 parsrar i analys av klinisk text, tillsammans med resultat som tidigare har uppnåtts för allmänspråket. Parsning av text är endast en del i ett eventuellt förenklingssystem, med fler steg som måste följa, men resultaten visar att existerande språkbearbetningsverktyg är fullt möjliga att använda efter en viss anpassning inom ramarna för ett sådant projekt.

Sett framåt kan den utarbetade programkedjan utvecklas genom träning av nya språkligt anpassade parsningsmodeller, vilket tillsammans med förbättrad förbearbetning och utökad guldstandard lär bädda för ännu bättre parsningsresultat. På sikt kan denna uppsättning förhoppningsvis bli en komponent i ett större förenklingssystem.

Litteraturförteckning

Aantaa, Kirsi. *Mot patientvänligare epikriser.* Masteruppsats; Nordiska språk; Åbo universitet, 2013.

Allvin, Helen. *Patientjournalen som genre.* Masteruppsats; Institutionen för nordiska språk; Stockholms universitet, 2010.

Ballesteros, Miguel & Nivre, Joakim. *MaltOptimizer: A System for MaltParser Optimization.* Proceedings of EACL'12 – Conference of the European Chapter of the Association for Computational Linguistics, 2012.

Buchholz, Sabine & Marsi, Erwin. *CoNLL-X shared task on multilingual dependency parsing.* CoNLL, 2006.

Anni Coden, Serguei Pakhomov, Rie Ando, Patrick Duffy, Christopher Chute. *Domain-specific language models and lexicons for tagging.* Journal of Biomedical Informatics, 2005.

Jung Fan, Elly Yang, Min Jiang, Rashmi Prasad, Richard Loomis, Daniel Zisook, Josh Denny, Hua Xu, Yang Huang. *Syntactic parsing of clinical text: guideline and corpus development with handling ill-formed sentences.* Am Med Info Ass, 2013.

Jeffrey Ferraro, Hal Daume, Scott Vall, Wendy Chapman, Henk Harkema, Peter Haug. *Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation.* Am Med Info Ass, 2013.

Peter Halacsy, Andras Kornai, Csaba Oravecz. *Hunpos – an open source trigram tagger.* Association for Computational Linguistics, 2007.

Martin Hassel, Aron Henriksson, Sumithra Velupillai. *Something Old, Something New – Applying a Pre-trained Parsing Model to Clinical Swedish.* Proceedings of NODALIDA'11 – Nordic Conference on Computational Linguistics, 2011.

Katri Haverinen, Filip Ginter, Veronika Laippala, Tapio Salakoski. *Parsing clinical Finnish: Experiments with rule-based and statistical dependency parsers.* Proceedings of NODALIDA'09 – Nordic Conference on Computational Linguistics, 2009.

Jurafsky, Daniel & Martin, James. *Speech and Language Processing.* Prentice Hall, 2008.

Kvist, Maria & Velupillai, Sumithra. *Professional Language in Swedish Radiology Reports – Characterization for Patient-Adapted Text Simplification.* Proceedings of SHI'13 – Scandinavian Conference on Health Informatics, 2013.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, Jan Hajič. *Non-projective dependency parsing using spanning tree algorithms.* Human Language Technology and Empirical Methods in Natural Language Processing, 2005.

Stephane Meystre, Guergana Savova, Karin Kipper-Schuler, John Hurdle. *Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research.* IMIA & Schattauer GmbH, 2008.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, Erwin Marsi. *MaltParser: A language-independent system for data-driven dependency parsing.* Natural Language Engineering, 2007.

Nivre, Joakim & McDonald, Ryan. *Integrating Graph-Based and Transition-Based Dependency Parsers.* Proceedings of ACL'08 – Human Language Technology, 2008.

Nivre, Joakim. *Svenska trädbanken.* Beskrivning av grammatiska funktioner; stp.lingfil.uu.se/~nivre/swedish_treebank [04.09.14]

Nivre, Joakim. *Algorithms for Deterministic Incremental Dependency Parsing.* Association for Computational Linguistics, 2008.

Skeppstedt, Maria. *Adapting a parser to clinical text by simple pre-processing rules.* Proceedings of BioNLP'13 – Workshop on Biomedical Natural Language Processing, 2013.

Smith, Kelly. *Treating a case of the mumbo jumbos: What linguistic features characterize Swedish electronic health records?* Masteruppsats; Institutionen för lingvistik och filologi; Uppsala universitet, 2014.

Östling, Robert. *Stagger: an Open-Source Part of Speech Tagger for Swedish.* Institutionen för lingvistik; Stockholms universitet, 2013.