Tallinna Ülikool

Informaatika Instituut

SPUM-

moodul dialoogisüsteemile DST asesõnade ja sünonüümide töötlemiseks

Seminaritöö

Autor: Stanislav Gaistruk

Juhendaja: Erika Matsak

Autor.....,.....,,2011

Juhendaja.....,.....,,2011

Instituudi direktor.....,.....,,2011

Tallinn 2010

Autorideklaratsioon

Deklareerin, et käesolev seminaritöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

04.01.2011 .....

(autor)

Sissejuhatus

Infotehnoloogia areng on toonud endaga kaasa lootusesädeme- võimaluse luua intelligents mis on võrreldav inimese omaga. Seda, kas huvi selle vastu tuleneb inimese soovist jumalat mängida või endast targem olevus luua, jääb juba psühholoogide ning filosoofide nuputada.

Üks oodatav omadus sellisel intelligentsi puhul on võime aru saada inimese kõnest ja/või kirjatekstist. Lisaks arusaamisele peaks selline intelligents suutma genererееrida inimese jaoks loomulikku, loetavat teksti. Nii arusaamine kui ka genereerimine on loomuliku teksti töötlemise (NLP – natural language processing) uurimisvaldkonnas (Paris, Cécile, Swartout, William R., Mann, William C.. Kluwer, 1991). Selles töös keskendume edaspidi ühele loomulikust tekstist arusaamisel tekkivale probleemile- sünonüümid ja asesõnad.

Erika Matsak on oma 2010 aastal valminud doktoritöö käigus täiustanud dialoogisüsteemi DST (Matsak 2005, 2009), mis töötleb tekstis olevaid sõnu neid esindavate predikaatloogika sümbolite abil (Lorents, 2000) ning teeb võimalikuks tuletussammude/-reeglite leidmise sisestatud tekstist. Probleemiks on erinevad sõnad tekstis mis tegelikult esindavad üht ja sama objekti. Praeguses lahenduses toimub nende asendamine käsitsi vahetult pärast teksti sisestamist. Oluline on, et sõnad mis on erinevad, kuid esindavad teksti või lause kontekstis üht ja sama objekti saaksid märgistusel koondatud ühe sümboli alla. Selleks on Erika Matsak loonud vastava sünonüüme ja asesõnu „ühtlustava“ mooduli loomiseks vajaliku algoritmi(Joonis 1). Seminaritöö peamiseks ülesandeks on algoritmi relaliseerimine ning selle realiseerimisel tekkivate probleemide analüüs.

Teema valikul motiveerisid varasem huvi tehisintellektiga seotu vastu (ka futuristlikud filmid ja spekulatsioonid) ning võimalus läbi seminaritöö protsessi saada aimu sellest kui lähedal on informaatika teadusena inimest korrektselt interpreteeriva intelligentse süsteemi loomisele. Samuti töö olemus- pikaaegse uurimistöö tulemusena valminud dialoogisüsteemi efektiivsuse tõstmine.

Seminaritöös antakse kõigepealt ülevaade moodulist ning tema vajalikkusest DST-le ja kasutusvõimalustest väljaspool DST-d. Tutvustatakse mooduli ülesehitust ning selle planeerimise/loomise käigus tekkinud probleeme mida ka analüüsitakse.

1. Moodulist SPUM (Synonym and Pronoun Unifying Module)

Selles peatükis vaatleme mooduli olulisust DST dialoogisüsteemile ning mooduli aluseks oleva metodoloogia kasutusvõimalusi mujal. Samuti üritame anda ülevaate mooduli loomisele eelnenud toimingutest.

1.1 DST ja SPUM-i vajalikkus DST-le

Moodul on vajalik selleks, et vähendada manuaalset sekkumist DST töösse vähendades sama tähendusega erinevate sõnade ja asesõnade arvu. DST praeguses versioonis toimub asesõnade ja sünonüümide töötlemine käsitsi, ning toob endaga kaasa suurenenud ajakulu ning vigade tekkimise tõenäosuse. Antud töö olulisus tuleb mängu suurte tekstide töötlemisel, kus paratamatult on suur sünonüümide ja asesõnade leidumise tõenäosus - sellest ka vajadus tekstide puhastamiseks asendades sõnade erinevad vormid ühe sobivaga kogu tekstis ning asendada asesõnad kindlate nimetajatega.

1.2 Võimalikud kasutusalad väljaspool DST-d

Kui väljuda DST kontekstist võib taolisele moodulile leida rakendusi erinevate loomulikku teksti töötlevate rakenduste puhul nende efektiivsuse tõstmiseks.

1.2.1 Otsingumootorid

Ebaeduka otsingu puhul kasutaja kas loobub otsingu teostamisest või muudab otsingu sätteid/otsisõna. Otsingu edu aga sõltub suuresti võtmesõnade valikust kasutaja poolt ning otsingusüsteemi võimekusest. Google patenteeris juba 2005 aastal meetodi, kus paralleelselt otsinguga kasutaja valitud võtmesõna järgi teostatakse seda ka sõnade sünonüümi järgi, parandades mõlemat: kasutaja valikut ning süsteemi võimekust (Lamping, J., & Baker, S., 2005). Google-i abil tehtavate otsingute hulka arvestades oleks selle meetodi rakendamise puhul kasulik lasta kasutajatel osaleda süsteemi õpetamises. Näiteks võib otsingumootorile luua lisana kasutajaga suhtlev osa, kus kasutaja võib peale otsingusõna sisestamist valida ka otsingumootori poolt pakutavatest potentsiaalsete sünonüümide seast sobivaima või sisestada see ise. Et kasutaja oleks nõus seda valikut teostama peaks süsteem vähendama potentsiaalsete sünonüümide arvu ning samal ajal suurendama nende hulgas olulisemate sünonüümide osahulka ja kasutaja(te) eelistuste kohta koguneva statistika alusel korrigeerima nende järjestust.

1.2.2 Ekspertsüsteemid

Nagu ka DST puhul oleks ka teadmistepõhise naturaalteksti töötleva ekspertsüsteemi töökuse tõstmisel kasulik tegeleda sünonüümide ning asesõnadega(Milward & Beveridge, 2003. Miltsakaki, 2010). Asesõnad mängivad siinkohal vähem rolli, kuna ekspertsüsteemi kasutajad on siiski teadlikud sellest, kuidas süsteemile end võimalikult arusaadavaks teha ja objektidele viidatakse korektselt. Juhul kui ekspertsüsteemidele leitakse laiemat kasutust väljaspool teadust ning kõrgtehnoloogiat- ühiskonna erinevates kihtides, tuleks tagada parem töökindlus ka ebamäärasema kasutajapoolse suhtluse korral.

1.3 Mooduli loomiseks vajalik eeltöö

Peale Erika Matsak poolt esitatud algoritmi(joonis 1) uurimist on algoritm modifitseeritud tööülesannete tükeldamise huvides. Peamine muudatus on algoritmi selge jaotus kaheks osaks: Pronoun mode(joonis 2.1) ja Synonym mode(joonis 2.2). See tõi kaasa parema ülevaate vajalikest klassidest ning projekti mahust. Samuti on lihtsam hinnata kogu töö vältel progressi ning välja tuua töö keerukamad osad.

Tulenevalt suurest päringute hulgast filosoft.ee veebilehele pidasin nõu Rene Prillopiga Filosoftist ning sain nõusoleku suuremamahuliste päringute tegemiseks.

2. Programmi kujundus

Siinkohal püüame anda ülevaate programmi GUI-st ning selgitada selle toimimist/funktsioone. Kuna tegu on mooduliga siis on oluline, et kasutajaliides oleks võimalikult lihtne ja kompaktne. Programmi töös soovimatute tulemuste ning vigade vältimiseks on võimalikku segadust tekitavad nupud välja lülitatud parajasti siis kui neid tarvis pole(Vt. joonis 3.1 ja 3.2.).

Infoväljade ja nuppude selgitused:

PRONOUNS- asesõnade režiim.

SYNONYMS- sünonüümide režiim.

Find pronouns- leiab sõnade hulgast asesõnad, infokonsool väljastab leitud asesõnade arvu.

Replace pronouns- alustab asesõnade asendamise protsessi.

Get word lemmas- leiab sõnadele võimalikud lemmad.

Unify lemmas- algatab lemmade ühtlustamise protsessi kus mitme võimaliku lemma puhul valib kasutaja välja kontekstile vastava.

Replace synonyms- alustab sünonüümide asendamise protsessi

Context- kuvab osa tekstist kus esineb sõna millele on tarvis valida õige lemma või asesõna asemel nimetaja.

Console- riba mille kaudu programm väljastab informatsiooni kasutajale.

3. Programmi struktuur

Selles peatükis selgitame GUI taga toimuvat ning struktuuri, mille poolest SPUM jaguneb programm kaheks: sünonüümide režiim ja asesõnade režiim. Kasutaja saab neid vahetada kui parasjagu ei toimu programmis päringute tegemist Filosofti.

3.1. Asesõnade režiim

Asesõnade režiimis toimub kasutaja algatusel asesõnade otsing mille käigus selgub millised sõnad tekstis kriteeriumile vastavad. Peale otsingu lõppemist võib alata asendamine, kus ükshaaval antakse kasutajale ette sõna ja osa tekstist kus sõna esineb ning kasutaja võib jätta sõna samaks või asendada ta kindla nimetajaga.

3.2. Sünonüümide režiim

Selles režiimis alustatakse kasutaja poolt sõnadele võimalike lemmade otsinguga, kus igale sõnale tekstist leitakse võimalikud lemmad. Kuna kontekstist

lähtuvalt saab sõnal olla ainult üks lemma siis järgmises etapis toimub lemmade ühtlustamine, kus kasutaja valib igale mitme võimaliku lemmaga sõnale ühe ja õige. Peale lemmade ühtlustamist võib alata lemmade sünonüümide otsing ning seejärel juba sünonüümide ühtlustamine, kus tekstis iga sõna puhul kontrollitakse, kas ta kuulub mõne tekstis eelpool esinenud sõna sünonüümide hulka.

3.3. Klassid

Programmeerimisel olen püüdnud järgida OO programmeerimise põhimõtet, kuid see õnnestus vaid selle tasemeni, kus erinevaid ülesandeid täitvad klassid tuli siduda kasutajaliidesega. Järgnevalt toon välja loodud klassid ja nende ülesanded:

Filosoft- päringute tegemine Filosofti pihta

Filter- Filosoftist saadud sisu puhastamine html koodist

Analyzer- filtreeritud info pakendamine ning analüüs

TextIO- lähteteksti failist lugemine, tulemuse väljastus

Slicer- lähteteksti tükeldamine sõnadeks ja kirjavahemärkide eemaldamine

SpumGUI- eelnevaid klasse siduv ja juhtiv peaklass, kasutajaliides

3.4 Lähtekood

Kogu lähtekood on kommenteeritud ning klassidele on loodud Java poolt pakutavate võimaluste abil(Eckel, 2002) dokumentatsioon ning on paigutatud CD-le(Lisad 4).

4. Teada olevad probleemid ning võimalikud lahendused

Antud peatükis toome välja programmi puuduste analüüsi tulemused ning mõningad programmeerimisel ilmnenud paradoksid. Järgnevalt välja toodud probleemid said ka sa

4.1. Kontekstiga mitteühilduvad sünonüümid

Kõige tõsisem probleem mooduli töös on sünonüümide asendamine juhtudel, kus ühe lemma kohta tagastab Filosoft õigustatut mitu gruppi sünonüüme. Näiteks võtame sõna *„algest“*, mille lemma *„alge“* on samaaegselt nimetavas ja omastavas käändevormis ning seega annab meile tulemuseks 2 gruppi sünonuume:

1. *„alge“* nimetavas käändes

   suge
   embrüo
   eos
   idu

2. *„alge“* osastavas käändes

   algusi
   hakatusi
   hakke
   alustusi
   sissejuhatusi
   alustamisi
   pihtahakkamisi
   pealehakkamis

Mooduli praeguses versioonis ei toimu mingisugust asendamist juhul kui tagastatakse rohkem kui üks grupp sünonüüme. Lahenduseks on siinkohal vastava algoritmi väljatöötamine, mille abil saaks kindlaks teha millisest grupist võrreldavale sõnale sünonüümi otsitakse või järjekordse dialoogi loomine kasutajaga. Liiga suure hulga dialoogide loomine on aga taunitav võttes arvesse mooduli ülesannet- võimalikult automatiseeritult puhastada tekst sünonüümidest ning asesõnadest.

4.2. Mitme võimaliku lemmaga sõnad

Raskuselt järgmine väljakutse on automatiseerida lemmade valik sõnale kui võimalikke valikuid on mitu. Siinkohal tundub ainuõige võimalus kontekstitundlik lemmatiseerimine ning see eeldab tohutute andmebaaside kasutusele võtmist statistika kogumiseks. Näiteks võtame sõna *„laud“*, mille 2 võimalikku lemmat on *„laud“*(nt köögilaud) ja *„laug“*(nt silmalaug). Sõna laud samaaegselt mitmuse nimetavas käändes ja ainsuse nimetavas käändes.

4.3. Asesõnade homonüümid

Ka asesõnade otsimisel tekstis pole võimalik 100% moodulile lootma jääda. Näiteks sõna *„keegi“* on esmamuljel selgelt asesõna, kuid sõnal pikemalt peatudes on olukord teine. Sõna keegi lemmad: *kee, keema, keegi*. Lahendusena on olemas programmil väljundriba kus kujutatakse osa tekstist milles sõna esineb, et kasutaja saaks programmi vea puhul sõna asendamise vahele jätta või asesõna puhul kontekstist lähtuvalt ta nimetajaga asendada.

5. Kokkuvõte

Seminaritöö tulemusena valminud programm kujutab endas prototüüp-moodulit Erika Matsak doktoritöö käigus täienenud dialoogisüsteemile DST (Matsak, 2009). See võimaldab töödelda moodulile ette antud tekstis olevaid asesõnu ning sünonüüme. Vaatamata prototüübi staatusele on mooduli praktikas kasutamiseks vajalike tegevuste olemus ja keerukus teada. Probleeme, mis mooduli rakendamist takistavad, on analüüsitud ning DST efektiivsuse tõstmine kasutades antud moodulit oleks lihtne, kuid selle rakendamise võimalused väljaspool DST-d vajavad põhjalikumat uurimist. Samuti vajaks põhjalikumat uurimist kasutaja sekkumise vajalikkuse vähendamise võimalused, mis on oluline, sest mooduli enda põhimõtteline eesmärk on vähendada kasutaja sekkumist DST töösse.

Töö praktiline pool valmis paralleelselt aine Programmeerimised I läbimisega, kust ammendatud oskused Java keeles programmeerimises tulid suuresti kasuks. Seminaritöö vältel omandatud teadmised ning arusaam antud informaatika harust on kindlasti suurenenud, kuid küsimusi sellel teemal tuli pigem juurde. Mõistes kui palju erinevaid viise võib olla ühe lause või koguni sõna tõlgendamiseks võib oletada, et loomulikku teksti töötleval informaatika harul on ees veel palju huvitavaid väljakutseid. Seda kinnitab kasvõi tõsiasi, et meil inimestena tekib omavahel suhtlemisel tihti arusaamatusi ning seda ka siis kui suheldakse n.ö ametlikus keeles.

Me ei mõista inimestena tihti teineteist ning ometigi on informaatikud valinud väljakutse luua süsteemi mis seda teeb - mõistab inimese kõne ning suudab seda ka genereerida. Tehisintellekti loomise katsetes on lähemale jõutud meie kujutelmale, mida viimaste aastakümnete jooksul mitmed kunstiinimesed on püüdnud väljendada ning loodan, et edusammud selles suunas jätkuvad.

Viited

Milward, D., & Beveridge, M. (2003). Ontology-based dialogue systems*. Proceedings of 3rd Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. *18th International Joint Conference on Artificial Intelligence.* London, UK. doi:10.1.1.136.5607. Retrieved from http://citeseerx.ist.psu.edu

Miltsakaki, E. (2010). *Antelogue: Pronoun Resolution for Text and Dialogue.* Coling 2010. Beijing, China. Loetud aadressil http://www.aclweb.org/anthology/C10-3011

Matsak, E. (2005*).* Dialogue System for Extracting Logic Constructions in Natural Language Texts. *The 2005 International Conference on Artificial Intelligence.* Las Vegas, USA.

Lorents, P. (2000). *Keel ja loogika*. Tallinn: EBS Print.

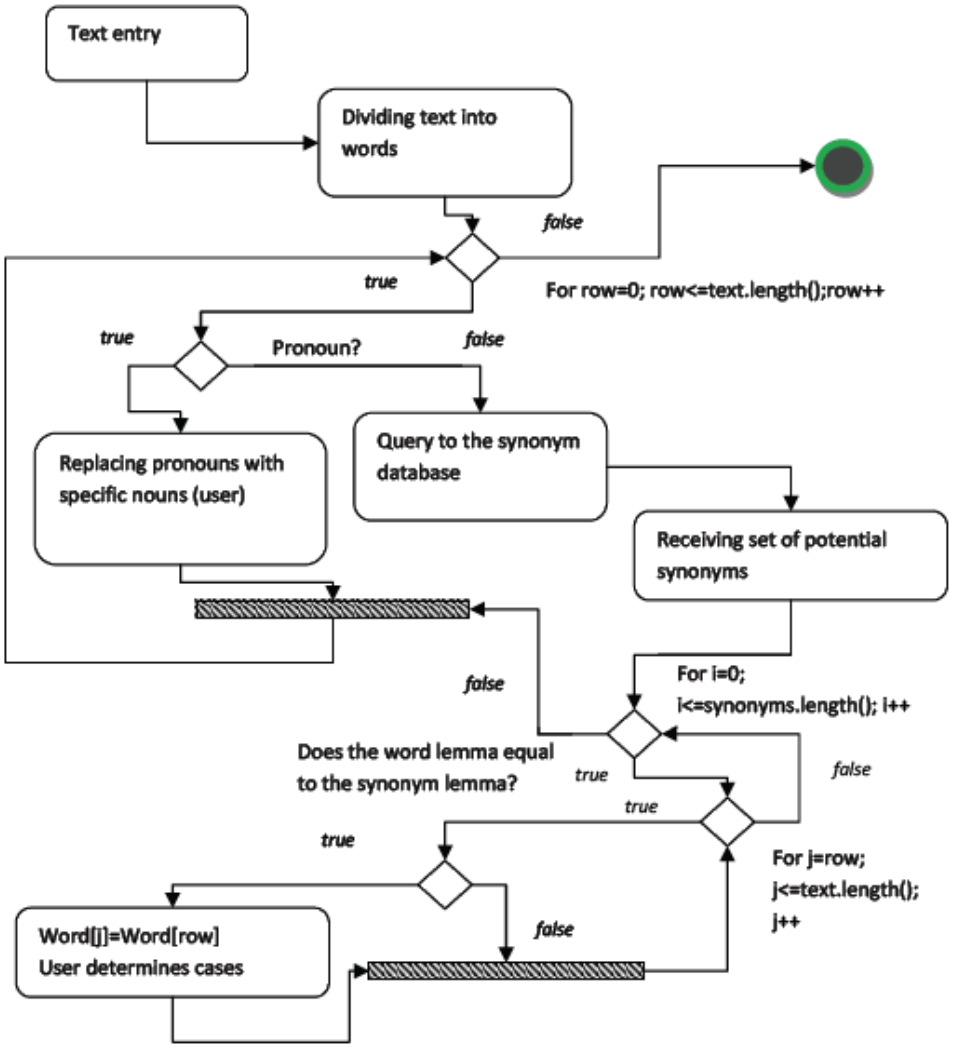
Paris, C., Swartout, W.R., & Mann, W.C. (1991). *Natural language generation in artificial intelligence and computational linquistics.* Cambridge, MA: MIT Press.

Matsak, Erika (2009). *Discovering Logical Constructs from Estonian Children Language.* Tallinn: TUT Press.

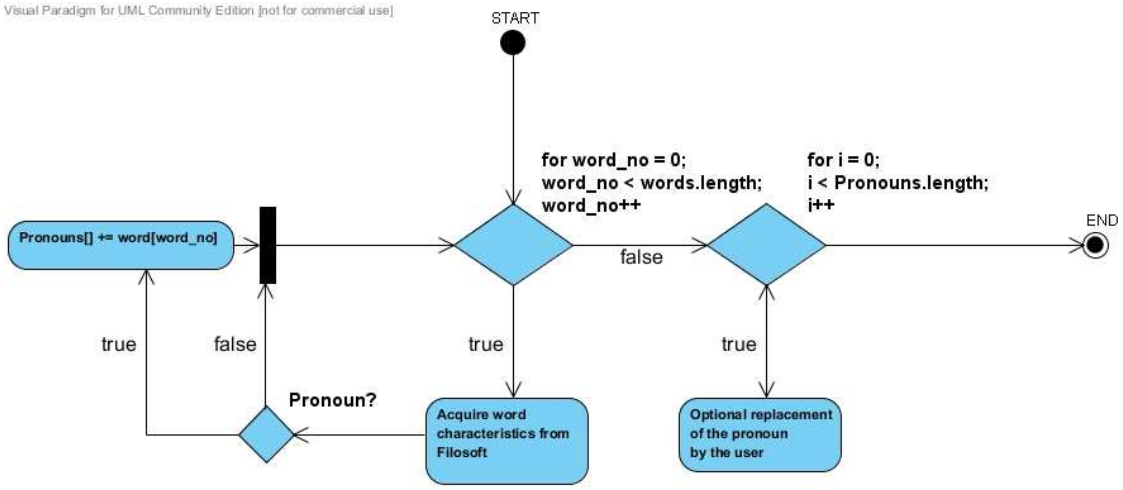
Eckel, B. (2002). *Thinking in Java (3rd Edition).* Upper Saddle River, NJ: Prentice Hall.

Lamping, J., & Baker, S. (2005). U.S. *Patent No.7636714.* Washington, DC: US Patent and Trademark Office. Loetud aadressil http://patft.uspto.gov/

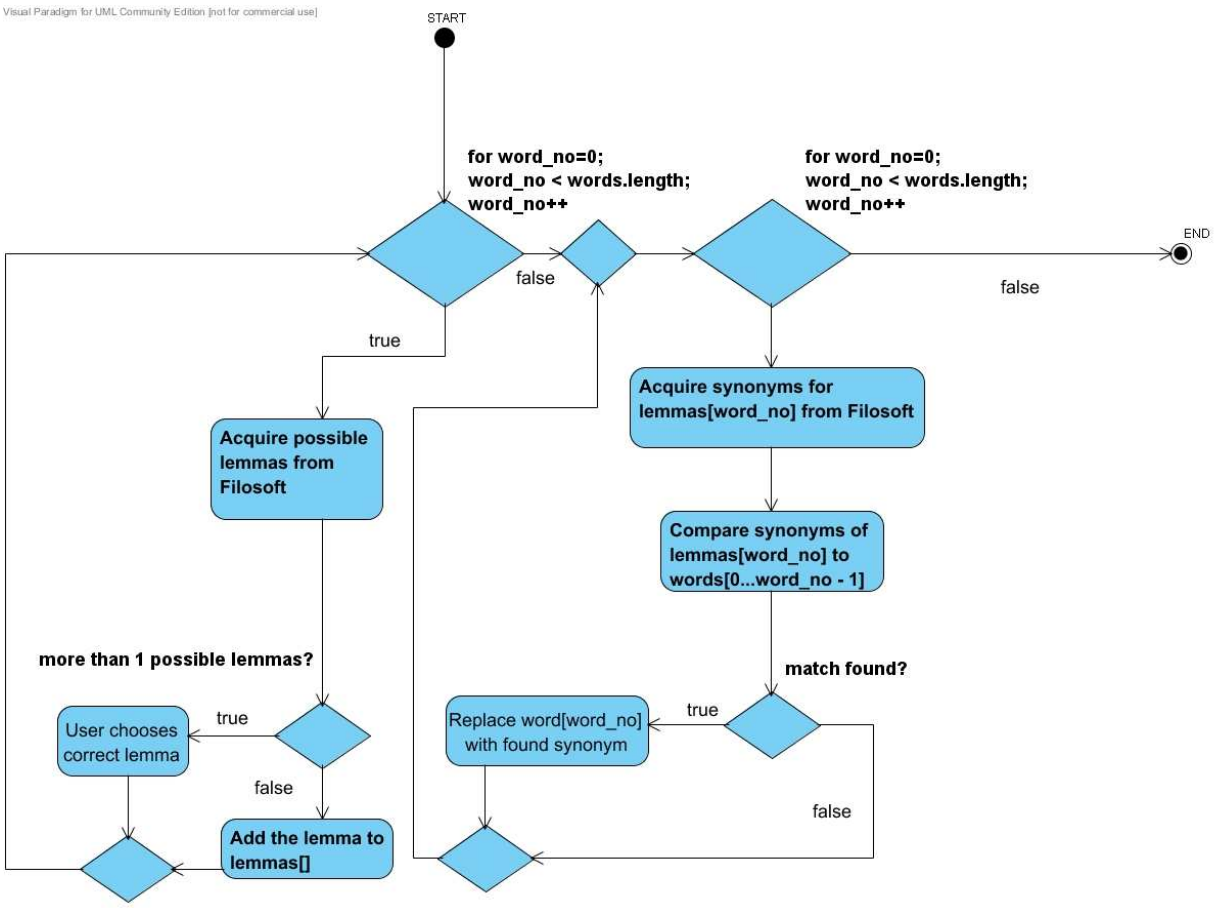
LISAD



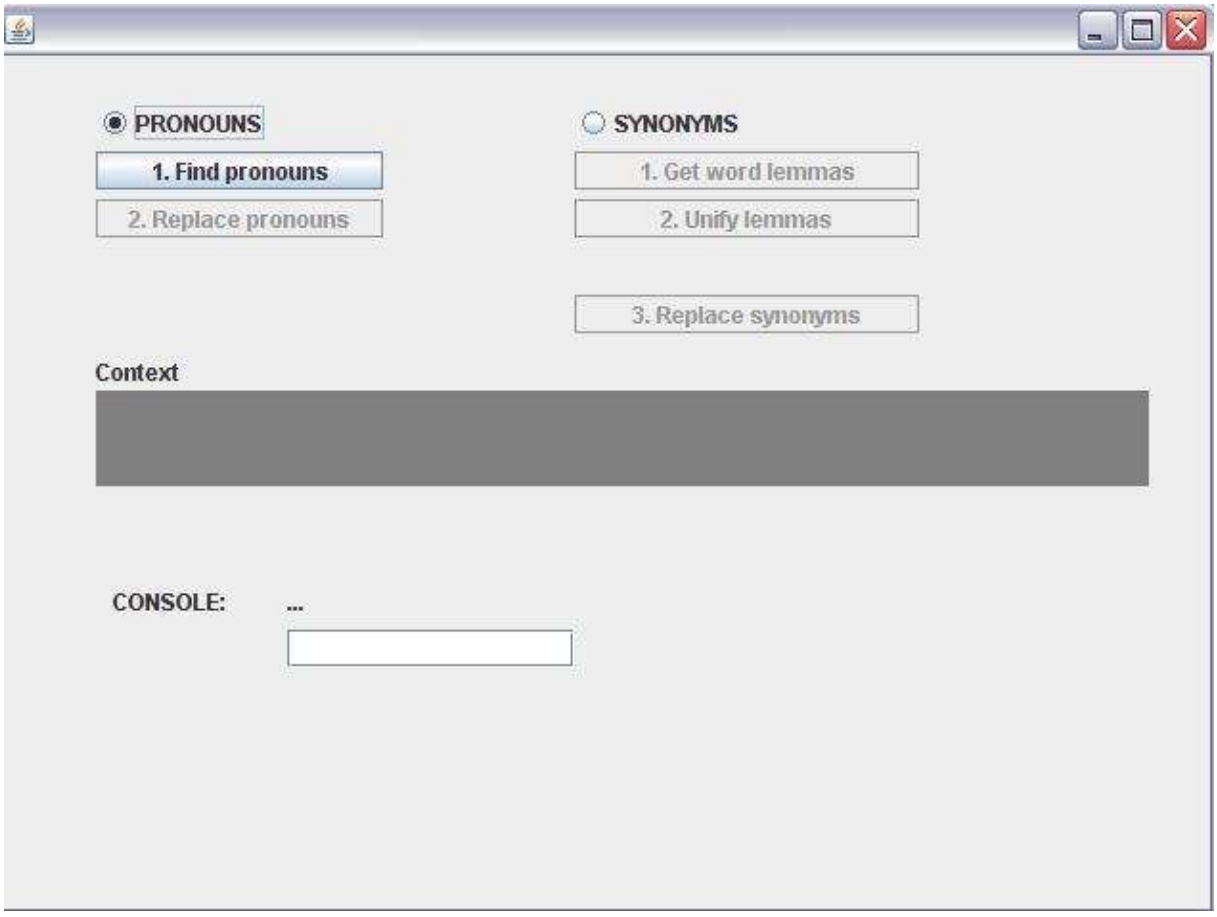
Joonis 1. Esialgne algoritm(Matsak, 2009)



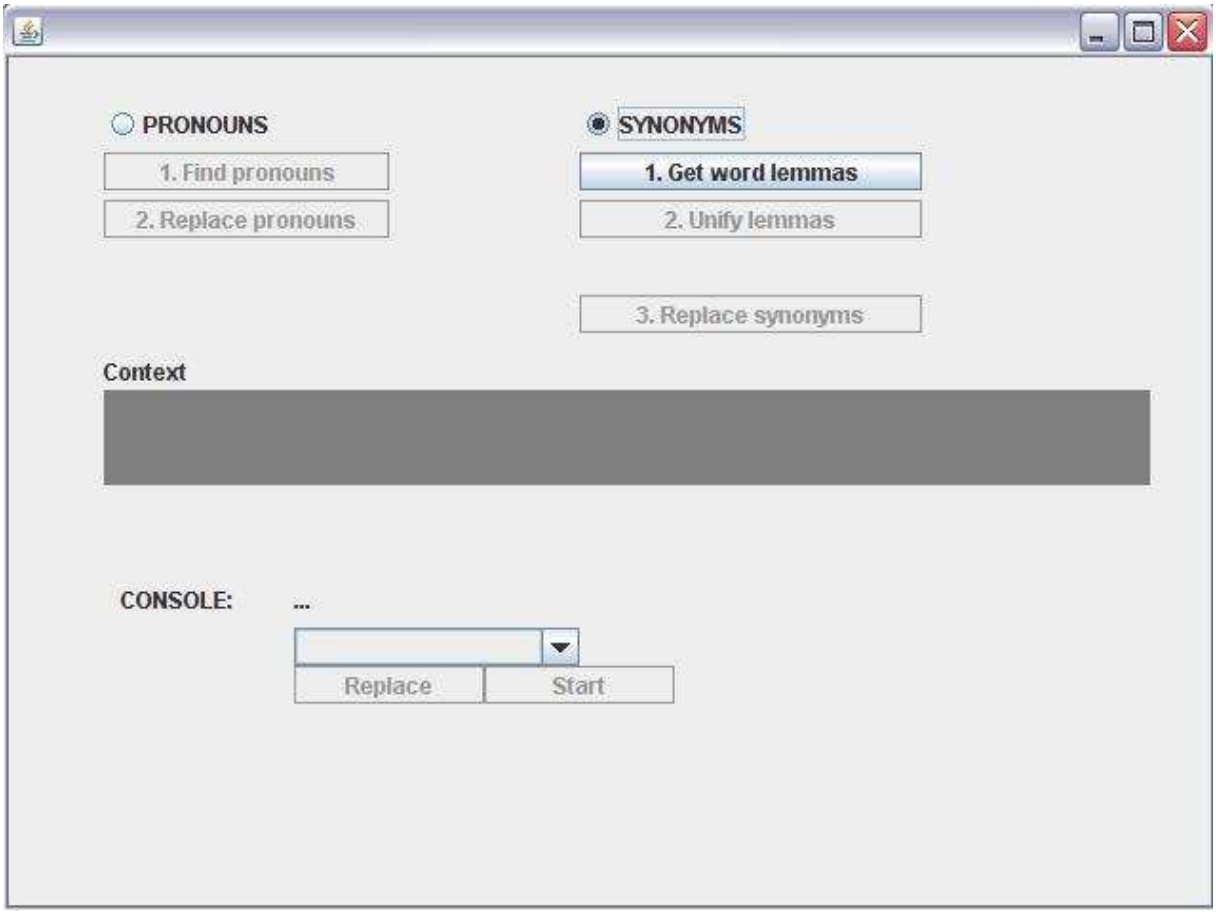
Joonis 2.1. Asesõnade asendamine



Joonis 2.2. Sünonüümide asendamine



Joonis 3.1. SPUM asesõnade töötelmise režiimi algolekus



Joonis 3.2. SPUM sünonüümide asendamise režiimi algolekus