Софийски университет Св. Климент Охридски

Факултет по славянски филологии

Магистърска програма по компютърна лингвистика и интернет технологии
в хуманитаристиката

ДИПЛОМНА РАБОТА

Машинен превод на граматичната категория определеност на нивото на именната фраза от шведски на български език, базиран на правила

Георги Илиев, факултетен номер 770342

Научен ръководител: доц. Светла Коева

София, 2009 г.

Съдържание

Увод

Настоящата работа се занимава с проблемите на машинния превод на именната фраза от шведски на български език. Изборът на тема бе продиктуван от професионалните интереси на автора и от теоретичната и практическа подготовка, получена по магистърската специалност «Компютърна лингвистика» в Софийския университет през академичната 2007/2008 г.

Работата е организирана в теоретична и практическа част. Теоретичната част започва с кратък преглед на съвременните технологии за машинен превод. Прави се проверка на функционалността на две общодостъпни системи, предлагащи превод от и/или на съответните езици. Въз основа на изводите от прегледа на работата на общодостъпните системи за машинен превод се взема решение за подхода към реализацията на практическата част. Целта на настоящата работа не е да се произнесе «за» или «против» даден метод за машинен превод за сметка на друг. По-скоро се търси действащ модел на система за машинен превод въз основа на наличните ресурси в мрежата понастоящем, който да бъде приложим на практика при превода между шведски и български език.

Частта «Езикови предпоставки за реализацията на NPTrans» представлява кратко въведение в морфологията на именната част на шведски език. Примерите, с които работи инструментът за машинен превод NPTrans, са достатъчно прости, за да не се налага владеене на шведски език за разбирането на функциите на NPTrans.

Въз основа на изводите от прегледа на морфологията на именната част на шведски и съпоставката с морфологията на именната част на български се съставя принципен модел за работата на инструмента за машинен превод, разделен на аналитичен и генеративен модул. Аналитичният модул работи на нивото на изходния език и неговата задача е да се отстрани многозначността на словоформите на нивото на именната фраза, да се провери дали именната фраза отговаря на зададените граматически правила и да се определят параметрите, общи за цялата именна фраза. Генеративният модул работи на нивото на целевия език върху корпуса, създаден в резултат от работата на аналитичния модул. Неговата задача е да генерира параметрите на отделните словоформи в състава на именната фраза на целевия език въз основа на общите параметри на именната фраза на изходния език и рода на съществителното на целевия език.

Функционалността на инструмента NPTrans се проверява с помощта на изходен корпус от прости примери за реализацията на именна фраза на изходния език. Архивът `NPTrans.rar` съдържа и анотиран корпус, съставен с помощта на аналитичния модул на NPTrans – всички именни фрази от статия от шведски ежедневник. С този корпус по удобен за потребителя начин (форматиране с XSLT) се илюстрира функционалността на аналитичния модул. Той служи за насока при разширяването / подобряването на логическия компонент, тъй като там ясно се вижда кои реализации на именната фраза от един реален текст се поддържат от NPTrans към момента и кои – не.

Понятието «машинен превод». Основни подходи към машинния превод

Машинният превод се състои в използването на компютърен софтуер за превод на реч или текст между различни естествени езици. В съвременните инструменти за машинен превод най-общо се прилагат два различни вида методи: емпирични[1] (*empirical*) (базирани на математически принципи) и лингвистични (*linguistic knowledge*) (базирани на граматически и/или онтологически правила). Изборът на подход към машинния превод зависи от специализацията на текста, наличните езикови данни, необходимата степен на граматична коректност на целевия текст и др.

Емпирични методи

Статистически машинен превод

В много случаи потребителят се нуждае най-вече от общия смисъл на представената информация на чужд език. За целта в мрежата често се използва инструмента *Google Translate*[10], включващ възможност за превод и между български и чужд език. Той представлява машина за превод, базиран на статистически методи. Най-общо работата на една такава система не зависи от езиците, между които се прави превод. Тя се основава на принципа на вероятността доколко даден езиков елемент (дума, фраза[2] или изречение) от целеви корпус е превод на съответен езиков елемент от изходен корпус[1]. При този метод не се налага кодиране на езикова информация – системата няма нужда да «знае» езиците, от/на които превежда.

Условие за функционирането на системата обаче е наличието на достатъчно голям двуезичен корпус – т.е. достатъчно количество вече преведен текст между съответната двойка езици. Освен това, за да може да служат на целите на статистическия превод, корпусите трябва да бъдат съотнесени (*aligned*) – на машината трябва предварително да се зададе съответствието между изреченията и/или думите в изходния и целевия корпус. При двойки езици с големи различия в словореда се налага да се посочи и позицията, в която се явява съответствието на всяка отделна дума в превода.

Примери за многоезични корпуси, послужили за основа на системи за статистически машинен превод, са протоколите *Hansard* от заседанията на канадския парламент, водени паралелно на английски и френски език[1] и съвкупността от нормативни

[1] Всички понятия на български, дадени в курсив на английски език, са превод на автора.
[2] Под фраза тук не се има предвид езикова фраза, а подредена последователност от символни низове.

документи на Европейския съюз *Acquis Communautaire* – паралелни текстове на 22 езика: български, чешки, датски, немски, гръцки, английски, испански, естонски, фински, френски, унгарски, италиански, литовски, латвийски, малтийски, нидерландски, полски, португалски, румънски, словашки, словенски и шведски. Голяма част текстовете в *Acquis Communautaire* са съотнесени в паралелен корпус под названието *JRC-Acquis* от Обединения изследователски център към Европейската комисия[16]. На базата на *JRC-Acquis* се реализира проекта SEE-ERA.net[2] за осигуряване на езикови ресурси и преводачески модели за машинен превод за южнославянските и балканските езици.

Редица изследвания показват, че качеството на изходните данни е от решаващо значение за статистическия машинен превод и то не може да бъде компенсирано с по-голям обем на многоезичните данни[2]. Тази констатация дава началото на проекта EuroMatrix[13], в който е възприет хибриден метод, съчетаващ предимствата на статистическия машинен превод с машинния превод, базиран на правила.

Машинен превод, базиран на примери (*Example-Based Machine Translation *или* EBMT*)

В системата за машинен превод, базиран на примери, се задава набор от изречения на изходния език и съответстващите им преводи на целевия език, които впоследствие се използват за превода на други подобни изречения от изходния на целевия език. В общия случай работата на *EBMT* не зависи от езиците, между които се превежда. Както и при статистическия превод, двуезичните корпуси, използвани при *EBMT*, също трябва да бъдат съотнесени по изречения и/или думи. Работата на системата се основава на допускането, че ако вече преведено изречение се срещне отново в текста, то вероятно същият превод отново би бил правилен[1].

Алгоритъм за машинен превод, базиран на примери, се прилага с определени ограничения в т. нар. преводаческа памет (*translation memory*) (у нас все по-често се използва *SDL Trados*). Преведените от преводач части от текста се добавят към базата от данни на преводаческата памет. Ако изходният текст съдържа изречение, което вече е въведено в базата от данни, неговият превод се вмъква в превеждания документ. Така на потребителя не се налага повторно да превежда същото изречение. Преводаческата памет е особено ефективна при превода на нова редакция на вече преведен документ (особено ако става дума за малки промени в новата редакция).

Преводаческата памет се интегрира в системи за управление на фирмена документация и способства за унифициране на терминологията, използвана в големи организации. Качеството на работа обаче зависи от това дали първоначално правилно е направен превод на съответната част от текста.

Прилагането на емпирични методи за машинен превод се основава на текст, произведен от преводач, който далеч не винаги е достатъчно компетентен за съответната област или дори за конкретния език, от/на който се превежда. Разликата в компетентността на преводачи, които работят върху отделни части на един и същ изходен текст, както и различните стандарти, прилагани от отделните преводачи, водят до различни интерпретации, а оттам и до по-ниско качество на превода, произвеждан от системи, в които се прилагат емпирични методи.

Езикови методи

*Машинен превод, базиран на правила (*Rule-Based Machine Translation*)*

Най-общо в машинния превод, базиран на правила, изходният текст се анализира до ниво на междинно символично представяне, от което може да се генерира текст на целевия език. В зависимост от вида на междинното представяне подходът се нарича или интерлингвален машинен превод (*interlingual machine translation*), или трансферен машинен превод (*transfer-based machine translation*). Анализът на изходния текст и генерирането на целевия текст става по предварително зададени системи от граматически и/или онтологически правила. За успешното прилагане на такъв метод се изисква наличието на богат речник с морфологична, синтактична и семантична информация и голям и по възможност изчерпателен набор от правила.

Машинният превод, базиран на правила, се основава на допускането, че е възможно да се зададе такова ниво на представяне на информацията, която носи текстът на изходния език, което е достатъчно абстрактно, за да може по недвусмислен начин да бъде преведено от машина, но същевременно е и достатъчно повърхностно, за да може синтактични конструкции на различни изходни и целеви езици по успешен начин да намерят съответствие на това ниво на абстрактно представяне[1].

На нивото на максимална абстракция всичката семантична и граматична информация, която носи изходният текст, се трансформира в елементи, независими от конкретния език. Това изцяло абстрактно ниво на представяне на езика се нарича интерлингва (*interlingua*). Оттам и методите за машинен превод, базиран на правила, които целят

постигане на максимална степен на абстракция на представянето, се наричат интерлингвални.

В трансферните системи за машинен превод, базиран на правила, се търси компромис между нивото на абстракция на представяне на текста на изходния език и компонента на системата, който зависи от конкретната двойка езици, между които става превода. Т.е. абстракцията на представяне на езиковата информация достига до определено ниво, на което елементи, които все още носят белезите на изходния език, се превръщат в елементи на целевия език. Целта на настоящата работа е реализацията на система за трансферен машинен превод, базиран на граматически правила.

Машинен превод, базиран на онтологически правила (*Knowledge-Based Machine Translation*)

В машинния превод, базиран на онтологически правила, освен описаните по-горе принципи на машинния превод, базиран на правила, при анализа на изходния текст и генерирането на целевия текст се въвежда и семантичен план. Т.е. системата, която извършва машинния превод, разполага с информация за семантиката и прагматиката в дадена област (домейн) и със средствата, необходими за вземане на решения до ограничена степен относно понятия в домейна и връзките между тях. Пример за такава система за машинен превод е KANT[8], която се разработва от 1989 г. насам от Центъра за машинен превод (CMU) към Университета Карнеги-Мелон. KANT намира успешно приложение в управлението на електроцентрали, при превод на техническа документация за тежкото машиностроене, воденето на медицински картони, превода на наръчници за автомобили и надписи за хора с увреден слух по телевизията.

Прототипните системи, разработени в края на 1980-те години в CMU, имат за цел автоматичен превод между английски и японски на наръчници за персонални компютри. Те съдържат следните компоненти:

- онтология на понятията, съставена от фреймове
- аналитичен компонент за лексиката и граматиката на английски и японски
- генеративен компонент за лексиката и граматиката на английски и японски
- правила за съответствие между абстракциите в интерлингвата и английския/японския синтаксис

Пример за фрейм от онтологията за понятието `computer`[1]:

Subclasses	personal-computer mini mainframe super
is-a	independent device
has-as-part	software computer-keyboard input-device disk-drive output-device CD-Rom card computer-hardware-card cpu memory-expansion-card monitor printer system unit
max-users	(<>1 200)
make	Plus AT XT 750 780
token	“The basic IBM Personal Computer consists of a system unit and keyboard”
Part-of	airport-check-in-facility security-check-device
operational	yes no
manufactured-by	intentional-agent
configuration	minimal regular extra
theme-of	device-event spatial-event

Домейнът се моделира по този начин с цел напълно отстраняване на многозначността, което води до много високо качество на целевия текст. Недостатъкът на системата е необходимостта от много подробно описание на понятията в домейна, което прави метода приложим само едновременно със строги ограничения за семантиката и граматиката в изходния текст, т.е. полезен е и дава високи резултати в много тясно специализирани области.

Примери за статистически машинен превод, машинен превод, базиран на примери, и машинен превод, базиран на правила

Статистически (*Google Translate*):

Превод на статия от *Euronews* от английски на български език,

<http://www.euronews.net/en/article/12/01/2009/russian-gas-taps-off-and-then-on-again/>

Изходен текст	**Целеви текст**
Gas Russian gas taps off and then on again (*Руският газ спира, а след това тръгва отново*)	Газ руски газ батерии разстояние, а след това отново
It is turning out to be the on-again-off-again deal to resolve the gas row between Russia and Ukraine.	Тя се превръща, за да бъде по-отново прихващане-пак сделката за разрешаване на газ ред между Русия и Украйна.
Gas monopoly Gazprom says Ukraine has now signed a new copy of an agreement over monitoring the transit of gas via Ukraine.	Газ монопол Газпром казва Украйна вече е подписано ново копие на споразумение по мониторинг на транзита на газ през Украйна.
Yesterday the Russian President said the deal was off because Kiev had added conditions. But now Moscow says the agreement has been signed without conditions. This is not the first time the deal has been in dealt, but fingers are crossed that full supplies can now be resumed.	Вчера на руския президент заяви, че сделката е на разстояние, защото Киев беше добавен условия. Но сега Москва казва споразумение е било подписано, без условия. Това не е първият път, когато сделката е била в разглеждат, но пръстите са преминали че пълното доставки може да бъде възобновено.
The presence of EU monitors in Ukraine is aimed at resolving a dispute that saw Russia cut off supplies because of what it said were unpaid bills.	В присъствието на наблюдатели на ЕС в Украйна е насочена към разрешаване на спора, който е видял Русия отсече доставки заради това, което той каза бяха неплатени сметки.
Moscow claimed that Ukraine then began stealing gas destined for Europe, to make up for its own shortfall. Kiev denied the accusations.	Москва заяви, че Украйна след което започва да краде газ, предназначени за Европа, за да се извърши за своя дефицит. Киев отрече обвиненията.
A fifth of the European Union's gas comes via Ukraine, and Brussels had little choice but to step in as mediator.	Една пета от Европейския съюз в областта на газ идва през Украйна, и в Брюксел имаше малък избор, освен да се оттегли като посредник.

Английският език доминира мрежата и е нормално качеството на статистическия превод от английски да е по-високо от това на превода от «по-малък» език, какъвто е шведският. *Google Translate* предлага дори и тази възможност. Пример отново по

темата, превод на статия от шведския ежедневник *Dagens Nyheter* от шведски на български език, <http://www.dn.se/DNet/jsp/polopoly.jsp?d=3130&a=873030>

Изходен текст	**Целеви текст**
Lösning på gaskrisen (*Решение на газовата криза*) Ett slutligt avtal har undertecknats av alla parter. Det sa Alexander Medvedev, vice vd för Gazprom, på en pressträff i Bryssel i dag. Ledande ryska regeringsföreträdare samt det ryska statliga gasbolaget Gazproms vd Aleksej Miller reste på måndagen till Bryssel för att där diskutera gaskrisen. På måndagen samlas också EU:s energiministrar till ett krismöte om gaskrisen där ett syfte uppges vara att inledningsvis hitta en snabb kortsiktig lösning på hur man bäst hjälper de länder vars befolkningar som drabbats av bråket. Senare på eftermiddagen meddelade Alexander Medvedev att de sista frågorna har rätats ut under mötet. Medvedev tror inte att några fler problem kommer att tillstöta, utan räknar med att Gazprom kan vrida på gaskranarna igen så snart inspektörer är på plats vid de överenskomna kontrollstationerna. - Om inga hinder uppstår kan gasleveranserna starta i morgon klockan åtta, säger Medvedev. Gasavtalet mellan Ukraina och det ryska statliga företaget Gazprom måste skrivas under av EU-kommissionen innan Ryssland kommer att återuppta leveranserna till Ukraina. Från TT	Разтвор на газ криза Окончателното споразумение е било подписано от всички страни. Това каза Александър Medvedev, заместник-главен изпълнителен директор на Газпром, на пресконференция днес в Брюксел. Водещият руски правителствени представители и Руската държавна газова компания Газпром изпълнителен директор Алексей Милър пътува в понеделник за Брюксел, за да обсъдят Газ криза. В понеделник, събрани на енергия в ЕС министри на спешно заседание на газовата криза, когато целта е да бъде отбелязан с първоначално намери бързо краткосрочно решение за това как най-добре да се помогне на страните, чиито популации засегнати от подред. Късно следобед Александър Medvedev обявява, че последната е rätats въпроси по време на срещата. Medvedev не вярвам, че повече проблеми, ще бъде да се изправи, но очаква, че Газпром може да се превърне в gaskranarna отново веднага след като инспектори са на мястото на договорените пунктове. - Ако няма пречка да започнете газови доставки на сутринта в осем часа, каза Medvedev. Gasavtalet между Украйна и Руската държавна компания Газпром трябва да бъде подписана от Европейската комисия, преди Русия ще възобнови доставките за Украйна. От TT

Прави впечатление повторението на грешката при превода на «спор» от английски език, където «*row*» – «спор, караница» и «*row*» – «редица» са омоними, и от шведски език, където съществителното «*bråk*» няма такова значение. И все пак, *Google Translate* дава и в двата случая погрешен превод на български език от едно и също семантично

гнездо: «газ ред» за «*gas row*» при превод от английски на български и «от подред» за «*av bråket*» при превода от шведски на български. Това навежда на мисълта, че инструментът *Google Translate* използва английския като междинен език (*pivot language*) при превода между езици, за които не е налице достатъчно голям двуезичен корпус – в случая шведски и български език (приблизително равен брой хора в света говорят шведски и български). Друго доказателство за това е и преводът на «*befolkningar*» – «населения» (думата не носи смисъла на «популация», за разлика от предполагаемото английско съответствие «*populations*»).

По-долу следва превод с *Google Translate* на първия абзац от «Пипи Дългото чорапче» на Астрид Линдгрен. Предмет на настоящата работа е машинният превод на именната фраза от шведски на български език, а описанието, с което започва «Пипи Дългото чорапче», е добър пример за това.

Изходен текст	**Целеви текст**
I utkanten av den lilla, lilla staden låg en gammal förfallen trädgård. I trädgården låg ett gammalt hus, och i huset bodde Pippi Långstrump. Hon var nio år, och hon bodde där alldeles ensam. Ingen mamma eller pappa hade hon, och det var egentligen rätt skönt, för på det viset fanns det ingen, som kunde säja till henne, att hon skulle gå och lägga sig, just när det var som roligast, och ingen som kunde tvinga henne att äta fiskleverolja, när hon hellre ville ha karameller.	На ръба на малък, малък град, е бил един стар изоставен градина. В градината е стара къща, а в къщата са живели Pippi Longstocking. Тя бе девет години и е живял там сам. Не майка или татко, тя е, и тя беше наистина хубаво така, защото така не е, което би могло да смеем да твърдим, да я, че ще отида в леглото, само когато тя е най-забавно, и няма кой да я принудят да се ядат риба, масло от черен дроб, когато тя е трябвало да сладкиши.

Признакът определеност на именната фраза «*den lilla staden*» – «малкия град» отсъства от превода на български. А при превода на «*en gammal förfallen trädgård*» – «стара запустяла градина» атрибутивната част на именната фраза не се съгласува по род със съществителното.

Машинен превод, базиран на примери – преводаческа памет (*ESTeam Translator*)

Извадка от система за автоматизиран превод на общоевропейска база от данни с информация за търговски марки:

Изходен текст – датски	**Целеви текст – български**
Design og udvikling af optaget computer software og af downloadable	Дизайнерски услуги и udvikling af optaget computer software и софтуер (компютърен) [с възможност за

computer software.	даунлоуд]

Трудно е да се прецени дали частите в скоби на български език са индивидуално решение на преводача или съответстват на вече преведени думи от изходния език в скоби. Става видно обаче, че при работа със система за машинен превод, базиран на примери, значение за качеството на целевия текст има не само качеството на превода в многоезичния корпус, а също и избора на минимална единица за превод (в случая това е думата). В конкретния пример заместването на «*Design*» с превода «дизайнерски услуги» не отговаря на синтактичната конструкция в изходния език.

Машинен превод, базиран на правила (*SYSTRAN*)

SYSTRAN е най-дълго разработваната система за машинен превод. Тя съществува от 1960-те години и успешно се е прилагала от Министерството на отбраната на САЩ за превод на руски технически документи по време на Студената война. *SYSTRAN* има внушителна история, но не предлага превод от/на български език. Поддържа следните езици: арабски, китайски, нидерландски, английски, френски, немски, гръцки, италиански, японски, корейски, полски, португалски, руски, испански и шведски[9].

Следва превод от шведски на английски език от сайта на *SYSTRAN* на статията <http://www.dn.se/DNet/jsp/polopoly.jsp?d=3130&a=873030>, преведена по-горе на български с *Google Translate*:

Изходен текст	**Целеви текст**
Lösning på gaskrisen Ett slutligt avtal har undertecknats av alla parter. Det sa Alexander Medvedev, vice vd för Gazprom, på en pressträff i Bryssel i dag. Ledande ryska regeringsföreträdare samt det ryska statliga gasbolaget Gazproms vd Aleksej Miller reste på måndagen till Bryssel för att där diskutera gaskrisen. På måndagen samlas också EU:s energiministrar till ett krismöte om gaskrisen där ett syfte uppges vara att inledningsvis hitta en snabb kortsiktig lösning på hur man bäst hjälper de länder vars befolkningar som drabbats av bråket.	Solution on the gas crisis A final agreement has been signed of all parties. It said Alexander Medvedev, vice CEO for Gazprom, on a pressure hit in Brussels today. Conductive Russian regeringsföreträdare and the the Russian State gas company Gazproms CEO Aleksej Miller traveled on Monday to Brussels in order to where to discuss the gas crisis. On Monday is collected also EU: s Ministers for Energy to a crisis meeting about the gas crisis where an aim be stated to last to initially to find it fast short-term lösnin […]

За безплатен достъп се прилага ограничение за дължината на текста, затова и статията не е преведена докрай. И все пак става видно, че анализът на нивото на фразата на

шведски не е достатъчно ефективен, тъй като се срещат повторения на определителния член (*the the Russian State gas company*), не е анализирана въобще формата за притежание «*Gazproms*», която съответства на английското «*Gazprom's*» и е налице повторение на глаголи в инфинитив (*to last to initially to find it*).

За проверка на генеративната функционалност на *SYSTRAN* при превода на именната фраза на шведски, който е с по-богата флективна система от английския, се използва част от друг популярен текст, «Питър Пан» на Дж. М. Бари:

Изходен текст	**Целеви текст**
Children have the strangest adventures without being troubled by them.	Barn har de konstigaste affärsföretagen, utan att besväras av dem.
For instance, they may remember to mention, a week after the event happened, that when they were in the wood they had met their dead father and had a game with him. It was in this casual way that Wendy one morning made a disquieting revelation. Some leaves of a tree had been found on the nursery floor, which certainly were not there when the children went to bed, and Mrs. Darling was puzzling over them when Wendy said with a tolerant smile:	För anföra som exempel, kan de minnas till omnämnande, en vecka efter händelsen händde, att, då de var i trät de hade mött deras döda fader och hade en lek med honom. Det var i detta tillfälligt långt att Wendy en morgon gjorde en oroande uppenbarelse. Något lämnar av en tree hade funnits på barnkammare däckar, som inte var bestämt där när barnen gick att bädda ned, och den Fru älsklingen förbryllade över dem, då Wendy sade med ett tolerantt leende:

Атрибутивните части и съществителните в именната фраза на шведски език са правилно съгласувани и правилно са генерирани признаците за членуване на именната фраза («*de konstigaste affärsföretagen*», «*deras döda fader*»). Прави впечатление обаче, че системата неправилно генерира формата за минало време на глагола «*att hända*», добавяйки окончанието *–de* към основа, завършваща на *–d*. Същото се отнася и за формата за род *neutrum* на прилагателното «*tolerant*». Може да се заключи, че *SYSTRAN* не прилага достатъчно прецизни правила за генерирането на флективни словоформи.

От представените примери може да се направи извод, че към настоящия момент няма широко достъпен инструмент за машинен превод, който да даде задоволителни резултати при превода между шведски и български език. Качеството на работа на наличните широко достъпни инструменти за машинен превод между други езици не е предмет на настоящата работа.

Формулировка на задачата. Избор на метод за машинен превод. Избор на среда за разработка

От краткия преглед на общодостъпните системи за машинен превод по-горе става ясно, че те не предлагат адекватна функционалност, що се отнася до превода между български и шведски език. Опитът с работата с някои от разпространените у нас системи за машинен превод (*ESTeam Translator*, *SDL Trados*) също не дава основание за оптимизъм, що се отнася до функционалността на комерсиалните инструменти за машинен превод между тези езици. Следователно, със създаването на инструмент за машинен превод между шведски и български език няма да се дублира функционалността на вече установен подобен инструмент.

Формулировка на задачата

От първостепенно значение за работата по един толкова обширен проблем, какъвто е машинният превод, е задаването на реалистични задачи за изпълнение. Освен това, с цел постигане на максимална полза от работата, крайният продукт следва да допуска надграждане или използване като обособен модул в състава на комплексна система за обработка и превод на текст.

От примерите, произведени от най-разпространените системи за автоматичен превод, става видно, че основната единица, от която успешно да се извлекат необходимите абстракции за генерирането на целевия текст, не може да бъде думата. За постигане на какъвто и да било резултат, различен от просто низ от несъгласувани речникови форми, е необходимо да се работи най-малко на нивото на фразата.

Съвременният шведски език и съвременният български език са до голяма степен аналитични езици, в които съществителното не се изменя по падеж в зависимост от ролята си в изречението или предлога, който го управлява. Така преводът на именната фраза не зависи от сказуемото или от предлога в състава на предложната фраза. Освен това съществуват неизменяеми части на речта с много висока честота, които при изпълнение на определени логически условия указват мястото на именната фраза в изречението – предлозите. Спецификата на шведския словоред допълнително спомага за точното определяне на местоположението на именната фраза в състава на простото изречение, било то в ролята на подлог, пряко или непряко допълнение, или рекция на предлог.

На пръв поглед именната фраза на шведски език изглежда най-подходяща за анализ на първите етапи от създаването на инструмент за машинен превод. Така първоначално бе формулирана и задачата на настоящата работа: създаване на инструмент за машинен превод на именната фраза от шведски на български език.[3]

Избор на метод за машинен превод

Най-важното условие за успешното прилагане на емпиричен метод за машинен превод е наличието на достатъчно голям обем двуезичен корпус с високо качество на превода. Такъв корпус от преводи между шведски и български език понастоящем липсва. Освен това, качеството на съществуващите преводи на нехудожествени текстове от скандинавски езици не е задоволително.

За прилагането на метод за машинен превод, базиран на правила, не се изисква предварително преведен текст. За сметка на това е необходимо да се работи с подробни речници с морфологична, синтактична и семантична информация за изходния и целевия език, които да позволят отстраняването на многозначност от изходния текст и правилния анализ на всички граматични и семантични параметри на словоформите в него, които впоследствие да бъдат преведени на абстрактно ниво в преводаческия софтуер и съответно да генерират целевия текст.

Такъв речник за шведски език съществува и се разпространява в мрежата с лицензa GNU LGPL от Езиковата банка към Гьотеборгския университет (*Språkbanken*, <http://spraakbanken.gu.se/>) като част от пакета *Saldo v.1.0*[7] (шведски асоциативен речник), разработен от Ларш Бурин, Маркус Форшберг и Ленарт Льонгрен и наличен за ползване в мрежата от 2008 г. Речникът на словоформите се генерира по правилата за словообразуване, зададени за лемите, включени в състава на основния речник към програмния продукт, и в компилиран вид съдържа общо 764375 словоформи (всички части на речта плюс съкращения, съществителни собствени имена, имена на художествени произведения и чуждици).

За български език такъв речник е речникът на словоформите DELAF, разпространяван със системата INTEX за български, която се използва за образователни цели в магистърската програма по компютърна лингвистика към Софийския университет. В

[3] В резултат от възникналите въпроси в хода на практическото изпълнение формулировката на задачата бе ограничена допълнително. Инструментът – предмет на настоящата работа предлага функционалността, описана в заглавието: машинен превод на признака определеност на нивото на именната фраза от шведски на български език, базиран на правила.

настоящия си вид речникът съдържа 12105 словоформи, част от които са добавени за превода на примерите, приведени в настоящата работа. Речникът на словоформите е част от електронния Граматичен речник на българския език. Той представлява списък от всички словоформи, съответната им основна форма и граматични характеристики. Граматичната информация в речника условно може да се раздели на категориална (граматични класове – съществително, прилагателно и т.н.), парадигматична (характеризираща основната форма и принадлежността на думата към съответния граматичен подклас – мъжки род; личен глагол, свършен вид и т.н.) и граматична (характеризираща образуването на словоформите). При въвеждането на категориалната информация е възприета традиционната класификация за частите на речта на български език (съществително, прилагателно, глагол, числително, местоимение, наречие, предлог, съюз, частица и междуметие), но граматичните класове, дефинирани в речника DELAF, не съвпадат изцяло с изброените части. В речника DELAF са дефинирани следните граматични класове: съществително име, прилагателно име, глагол, числително име, лично местоимение, притежателно местоимение[4], възвратно местоимение, местоимение, неизменяема част[3].

Наличието на тези ресурси бе сериозно основание за прилагането на метод за машинен превод, базиран на правила, при разработката на инструмента – предмет на настоящата работа.

Избор на среда за разработка

Сред основните принципи при разработката на инструмента – предмет на настоящата работа бе ограничаването на строго техническите аспекти до минимум с цел по-опростено и стройно представяне на методите, които се прилагат за постигане на поставените цели. Доколкото е възможно, програмирането става на високо ниво и за изпълнението на стандартни действия се прилагат готови библиотеки, разпространявани като част от пакета *ActivePerl 5.10.2*.

Основната част от инструмента е написана на Perl. Perl е изключително гъвкава среда за разработка на CGI скриптове за обработка и генериране на динамично съдържание в интернет и не зависи от платформата, върху която работи програмата. С някои малки модификации представеният с настоящата работа инструмент може да се превърне в система за превод онлайн с графичен потребителски интерфейс. Не на последно място,

[4] Вж. бележка под линия 18 на с. 50 по-долу

Perl предлага много добри възможности за включване на модули, които трудно биха се интегрирали в други широко разпространени езици за програмиране (в случая модула за логическо програмиране на Prolog `AI::Prolog`).

Корпусът, върху който инструментът работи, е във формат XML. XML е сред най-широко използваните методи за представяне на данни в мрежата понастоящем. Той осигурява приемственост между програмни продукти от различни поколения и използването му би улеснило разширението на настоящия инструмент и интегрирането му в комплексна система занапред. Освен това, XML е форматът, възприет от консорциума TEI[12] като стандарт за представяне на текстове в цифровизиран вид. Въпреки че в настоящия си вид корпусът, върху който работи инструмента – предмет на настоящата работа, не е съобразен с препоръките на TEI, той лесно може да се съвмести с тях.

Аналитичните и генеративните функции на инструмента (неговата «граматика») са изцяло обособени в компоненти, написани на Prolog. А това позволява допълнителното прецизиране на граматиката и нейното разширение да става независимо от чисто програмния компонент на инструмента, написан на Perl. Структурата на програмата на Prolog (която, поради спецификата на езика и поради факта, че той силно се различава от т. нар. процедурни езици за програмиране, често се нарича «база от данни» или просто «текст») се състои от факти (формални взаимоотношения между обекти) и правила за проверка на това дали даден факт може да бъде извлечен от базата от данни. При изпълнение на програмата интерпретаторът на Prolog претърсва базата от данни в опит да установи дали въпросният факт може да бъде потвърден въз основа на фактите и правилата в нея. Фактът, който бива потвърден или отхвърлен в резултат от работата на програмата, се нарича «цел»[11]. Prolog е логически език за програмиране от високо ниво, чиято употреба най-често се свързва с изкуствения интелект и компютърната лингвистика.

Речникът на словоформите на шведски от пакета *Saldo v1.0* може да бъде генериран в един от следните три формата:

- JSON (*JavaScript Object Notation*)
- XML
- SQL (по-точно поредица от заявки към база от данни, с които в таблица от базата се записват редове със съответната езикова информация)

Неписано правило при представяне на езикови данни е използването на маркиращ език или такъв, който позволява четене не само от машини. Направиха се опити с търсене и извличане на информация от речника на словоформите във всеки от трите формата, като базата от данни и в трите случая е от порядъка на 90 MB. При работата с речника на словоформите във формата JSON и XML търсенето става в пъти по-бавно, отколкото при търсене в база от данни със заявки на SQL.[5] Затова и в окончателния си вид инструментът за машинен превод на именната фраза – предмет на настоящата работа използва търсене в речника на словоформите в база от данни MySQL.

Всички от използваните софтуерни компоненти за разработката на инструмента – предмет на настоящата работа[6] се предлагат във вид за безплатно ползване по съответните лицензи:

- ActivePerl (включително използваните библиотеки `XML::Twig`, `AI::Prolog` и `DBI`) по лиценза ActiveState Community License
- MySQL Community Server по лиценза GNU GPL
- Saldo v.1.0 по лиценза GNU LGPL

[5] С прилаганите от автора алгоритми за търсене. Вероятно има неизследвани възможности за оптимизация на търсенето за форматите JSON и XML, които не са предмет на настоящата работа.
[6] В описанието на реализацията на инструмента – предмет на настоящата работа за удобство той условно ще се нарича NPTrans.

Езикови предпоставки за реализацията на NPTrans

Категориите *genus* (род), *numerus* (число), *kasus* (падеж) и *species* (определеност) в шведския[4]

Родът на съществителното на шведски език, подобно на рода на съществителното на български, е лексикално-граматична категория[6], която, наред с категориите число, падеж и определеност, определя формалните характеристики на именната фраза. В съвременния шведски език съществителните се поделят на два рода, т. нар. *utrum* и *neutrum* или *en* и *ett* род, където *en* и *ett* са съответните предпоставени неопределителни членове в единствено число на съществителните, а *–(e)n* и *–(e)t* са определителните членове, които се поставят в постпозиция (окончания) за членуване на съществителното в отсъствието на определения. Определителният член в постпозиция е особеност на всички скандинавски езици[5].

български	*шведски*	*български*	*шведски*
къща	(**ett**) hus	книга	(**en**) bok
къща**та**	hus**et**	книга**та**	bok**en**

При наличие на прилагателно в атрибутивна употреба обаче членуването на именната фраза става с помощта на предпоставен определителен или неопределителен член (съответно *den* за съществителни от род *utrum* и *det* за съществителни от род *neutrum*).

български	*шведски*	*български*	*шведски*
голям**а** къща	(ett) stor**t** hus	дебел**а** книга	(**en**) tjock bok
голяма**та** къща	**det** stor**a** hus**et**	дебела**та** книга	**den** tjock**a** bok**en**

Подобно на прилагателното на български, прилагателното на шведски език се съгласува по род и число със съществителното, което определя. В единствено число в неопределена форма това означава, че прилагателното приема окончание *–t*, ако определя съществително от род *neutrum*. В единствено число в неопределена форма прилагателното не приема окончание, ако определя съществително от род *utrum*. Но за разлика от прилагателното на български, прилагателното на шведски се съгласува и по

признак «определеност» със съществителното. Това означава, че в определена форма прилагателното приема общо окончание *–а* за двата рода.[7]

български		*шведски*	
зелен балон (м. р.)	зелени**я(т)** балон	(**en**) tjock bok	**den** tjock**a** bok**en**
голям**а** къща (ж. р.)	голям**ата** къща	(**ett**) stor**t** hus	**det** stor**a** hus**et**
страшн**о** куче (ср. р.)	страшн**ото** куче		

Докато на български в състава на членуваната именна фраза от горния вид маркер за признака «определеност» запазва само прилагателното (или по-точно само първото определение в случай на конструкция от вида «големи**я(т)** зелен балон»), то на шведски се наблюдава т. нар. тройно (редундантно) членуване с предпоставен определителен член, маркер за определеност на прилагателното и определителен член на съществителното.

Подобно на български, формата на прилагателното за множествено число няма маркер за род. На шведски прилагателното в множествено число получава общо окончание *–а*.

български – мн. ч.		*шведски – мн. ч.*	
големи балони	**големите** балони	stor**a** böcker	**de** stor**a** böcker**na**
големи къщи	**големите** къщи	stor**a** hus	**de** stor**a** hus**en**
големи кучета	**големите** кучета		

Редундантното членуване се запазва и в множествено число, където предпоставеният определителен член е общ за двата рода (*de*). В отсъствието на определения определената форма в множествено число се образува с помощта на определителен член в постпозиция.

български – мн. ч.		*шведски – мн. ч.*	
къщи	къщи**те**	hus	hus**en**

[7] Или окончание *–е* за лица в мъжки род единствено число («*den store mannen*» – «*големия(т) мъж*»). В миналото на шведски е имало три рода и тази употреба е останка от мъжки род. Тя обаче не е задължителна[5].

книги	книги**те**	böcker	böcker**na**

Съществителните в съвременния шведски имат две падежни форми: основна форма и форма за родителен падеж (притежателна форма; генитив). Маркерът за родителен падеж на съществителните е окончанието –*s*. За съществителните в определена форма то се поставя след определителния член.

основна форма	*притежателна форма*
Peter har bil.	Peter**s** bil
Петер има кола.	колата на Петер

За разлика от шведския, на български език притежателната форма най-често е аналитична и в превод от шведски на български тя има фертилност (*fertility*) > 1[1]. Това означава, че словоформата на изходния език съответства на повече от една словоформа на целевия език. Това поражда допълнителни усложнения в машинния превод и към момента NPTrans не съдържа механизъм за генериране на този вид конструкции.

Особеност на притежателните форми на шведския език е, че при тях прилагателното в атрибутивна функция следва правилата за образуване на членувана именна фраза.

основна форма	*притежателна форма*
Peter har en röd bil.	Peter**s** röd**a** bil
Петер има червена кола.	червена**та** кола на Петер

От представения пример става видно, че в превод на български прилагателното също се членува, ако изпълнява атрибутивна функция в състава на притежателна форма.

Аналогично се образуват и формите с притежателни местоимения на шведски.

основна форма	*притежателна форма*
Jag har en röd bil.	**min** röd**a** bil
Имам червена кола.	моята червена кола
Jag har två röda bilar.	**mina** röd**a** bilar
Имам две червени коли.	моите червени коли

Част от притежателните местоимения също се съгласуват по род и число със съществителното.

utrum			
min bil	моята кола	min**a** bilar	моите коли
din bil	твоята кола	din**a** bilar	твоите коли
hans/hennes/dess bil	неговата / нейната / неговата кола	hans / hennes / dess bilar	неговите / нейните / неговите коли
vår bil	нашата кола	vår**a** bilar	нашите коли
er bil	вашата кола	er**a** bilar	вашите коли
deras bil	тяхната кола	deras bilar	техните коли
neutrum			
mi**tt** hus	моята къща	min**a** hus	моите къщи
di**tt** hus	твоята къща	din**a** hus	твоите къщи
hans/hennes/dess hus	неговата / нейната / неговата къща	hans / hennes / dess hus	неговите / нейните / техните къщи
vår**t** hus	нашата къща	vår**a** hus	нашите къщи
er**t** hus	вашата къща	er**a** hus	вашите къщи
deras hus	тяхната къща	deras hus	техните къщи

Притежателните местоимения на шведски имат и възвратна форма, която също се съгласува по род и число със съществителното.

hans hus	неговата къща	sitt hus	своята къща
hans hus	неговите къщи	sina hus	своите къщи

В трансферната част на NPTrans формите с притежателно местоимение на шведски език следва да се превръщат в членувани форми с притежателно местоимение на български език.

Изводи за функциите, които следва да изпълнява инструментът за машинен превод NPTrans

Вид на именната фраза

Настоящата работа трудно би могла да обхване всички възможни реализации на именната фраза на шведски език. По-скоро нейната задача е да реализира действащ на практика модел за машинен превод на именната фраза с възможност за усъвършенстване на логическия компонент отделно от останалите компоненти.[8] Зададени са следните ограничения:

- Именната фраза се разглежда като едно цяло, съставено от равнопоставени елементи (словоформи). От една страна така се постига по-голяма степен на универсалност на инструмента. От друга обаче по този начин е трудно да се зададе подходящ модел на аналитичните форми за степенуване на прилагателните. Един по-специален случай е модификацията на прилагателното с наречие от вида «*ett mycket stort hus*» – «много голяма къща». Логическият модел на тази фраза е по-ясен, тъй като наречието е неизменяема част на речта, за чиито превод не се налага анализ на фразата на нивото на изходния език.[9] За улеснение и за по-ясно илюстриране на модела, прилаган в работата на NPTrans, няма да се задават логически модели за именни фрази с прилагателно в сравнителна или превъзходна степен с атрибутивна функция (нито синтетични, нито аналитични форми). Това означава, че фрази от рода на «*ett mer demokratiskt samhälle*» – «по-демократично общество» няма да бъдат разпознавани като валидни шведски фрази с възможност за машинен превод.
- Освен от прилагателно, атрибутивна функция в именната фраза може да се изпълнява и от:
  - предложна фраза: «*en klänning* ***med gula blommor***» – «рокля с жълти цветя»;
  - подчинено изречение: «*en klänning*, ***som är fin***» – «рокля, която е хубава»;

[8] Моделът се оказа успешен още при подготовката на настоящата работа. В първоначалния си вид NPTrans поддържаше много малък брой варианти на именната фраза на изходния език. За кратко този брой бе разширен почти двойно, което се постигна почти изключително с работа само по логическия компонент.

[9] Вероятно би било по-уместно в този случай атрибутивната част на именната фраза да се разглежда като цяло. Това би било решение също и на проблема с аналитичните форми за степенуване на прилагателното.

- инфинитивна фраза: «*en klänning* ***att vara fin i***» – «рокля, в която да бъде хубава».

В първия случай предложната фраза може да се анализира рекурсивно и рекцията на предлога да се преведе като самостоятелна именна фраза. Останалите два случая не са предмет на настоящата работа.

Целта на NPTrans на нивото на изходния език ще бъде да разпознава като валидни именни фрази на шведски с възможност за машинен превод следните възможни реализации:

bil	кола	de tre bilarna	трите коли
bilar	коли	de tre svarta bilarna	трите черни коли
bilen	колата	de tre alldeles svarta bilarna	трите напълно черни коли
bilarna	колите	svarta tavlan[10]	черната дъска
en bil	кола	det höga huset	високата къща
fint väder	хубаво време	de höga husen	високите къщи
en fin blomma	хубаво цвете	en mycket god macka	много вкусен сандвич
fina blommor	хубави цветя	mycket goda mackor	много вкусни сандвичи
en blomma	(едно) цвете	den mycket goda mackan	много вкусния(т) сандвич
en vacker blomma	(едно) хубаво цвете	de mycket goda mackorna	много вкусните сандвичи
en mycket vacker blomma	(едно) много хубаво цвете	hennes (sin) klänning	нейната (своята) рокля

[10] Конструкции от този вид представляват изключение от правилото за членуване на съществително, определено от прилагателно с атрибутивна функция. Те биват няколко вида: клиширани изрази (като настоящия пример), географски понятия и имена на институции (напр. «*Svarta havet*» – «Черно море» или «*Svenska institutet*» – «Шведския институт»), прилагателни, които семантично предполагат опозиция или конкретна алтернатива (напр. «*högra handen*» – «дясна ръка»), и прилагателните «*hel*», «*halv*» и «*själv*»[4]. Тъй като критериите за това какво може да се възприема като клиширан израз са неясни и варират (напр. «*svenska folket*» – «шведския(т) народ» е клиширан израз и граматически коректен без определителен член, но «**bulgariska folket*» – «българския(т) народ» не е и изисква определителен член), по-уместно би било в система за машинен превод те да се задават отделно като изключения, собствени имена, топоними и т.н.

tre blommor	три цветя	hennes (sin) fina klänning	нейната (своята) хубава рокля
tre vackra blommor	три хубави цветя	hans (sina) skor	неговите (своите) обувки
tre mycket vackra blommor	три много хубави цветя	hans (sina) bruna skor	неговите (своите) кафяви обувки

Род на съществителното

Лексикално-граматичната категория «род» на шведски в диахронен аспект се е развила по различен начин от същата категория в славянските езици. Трите рода (*maskulinum*, *femininum* и *neutrum*) в старошведския са се преобразували в съвременния шведски език в два (*utrum* и *neutrum*) с някои останки, като в групата на род *utrum* най-общо попадат съществителните от род *femininum* и *maskulinum* от старошведския[5]. Това само по себе си прави невъзможен паралела между рода на съществителното на български (мъжки, женски и среден) и рода на съществителното на шведски в синхронен аспект. Съществителни от двата рода на шведски се превеждат със съществителни от трите рода на български.

utrum		*neutrum*	
en häst	кон (м. р.)	ett skepp	кораб (м. р.)
en bil	кола (ж. р.)	ett äpple	ябълка (ж. р.)
en växt	растение (ср. р.)	ett barn	дете (ср. р.)

От посочените примери можем да заключим, че родът на съществителното в изходния език не би следвало да се предава в трансферната част на NPTrans. Той обаче ще служи за основа при проверка на фразата на изходния език.

Граматична омонимия на окончанието –a на прилагателното име на шведски

Прилагателното име на шведски приема окончание *–a* във всички случаи, когато изпълнява атрибутивна функция в състава на именна фраза, освен ако съществителното не е нечленувано в единствено число. Така граматичната омонимия на формата на прилагателното с маркера *–a* е следната:

*stor**a***

stor**a** hus	(*ett* род, мн. ч. нечл.)	голем**и** къщи
det stor**a** huset	(*ett* род, ед. ч. чл.)	голям**ата** къща
de stor**a** husen	(*ett* род, мн. ч. чл.)	голем**ите** къщи
stor**a** hundar	(*en* род, мн. ч. нечл.)	голем**и** кучета
den stor**a** hunden	(*en* род, ед. ч. чл.)	голям**ото** куче
de stor**a** hundarna	(*en* род, мн. ч. чл.)	голем**ите** кучета
mitt stor**a** hus	(*ett* род, ед. ч. след притежателна форма)	моята голям**а** къща
mina stor**a** hus	(*ett* род, мн. ч. след притежателна форма)	моите голем**и** къщи
min stor**a** hund	(*en* род, ед. ч. след притежателна форма)	моето голям**о** куче
mina stor**a** hundar	(*en* род, мн. ч. след притежателна форма)	моите голем**и** кучета

Очевидно фактът, че прилагателното в изходния език носи маркера *–a*, не дава почти никаква информация за формата на прилагателното в повърхностната структура на фразата на целевия език. А както видяхме в предходния раздел, маркерът за род в единствено число на изходния език също не носи такава информация.

Членуване и число на именната фраза

Независимо от различията в реализацията на категорията «определеност» в повърхностната структура на именната фраза на шведски и на български език, тя се запазва. Т.е. членуваната именна фраза на шведски език се превежда с членувана именна фраза на български език. На български се прави разлика между пълен и кратък член в зависимост от ролята на именната фраза в мъжки род единствено число в изречението, но на шведски такава разлика няма. Синтактичният анализ на нивото на изречението не е предмет на настоящата работа, затова логическият компонент на NPTrans не следва да прави разлика между кратък и пълен член на български език.

Категорията «число» на фразата на целевия език също се запазва при превода. Т.е. именната фраза на шведски език в множествено число се превежда с именна фраза в множествено число на български език и обратно.

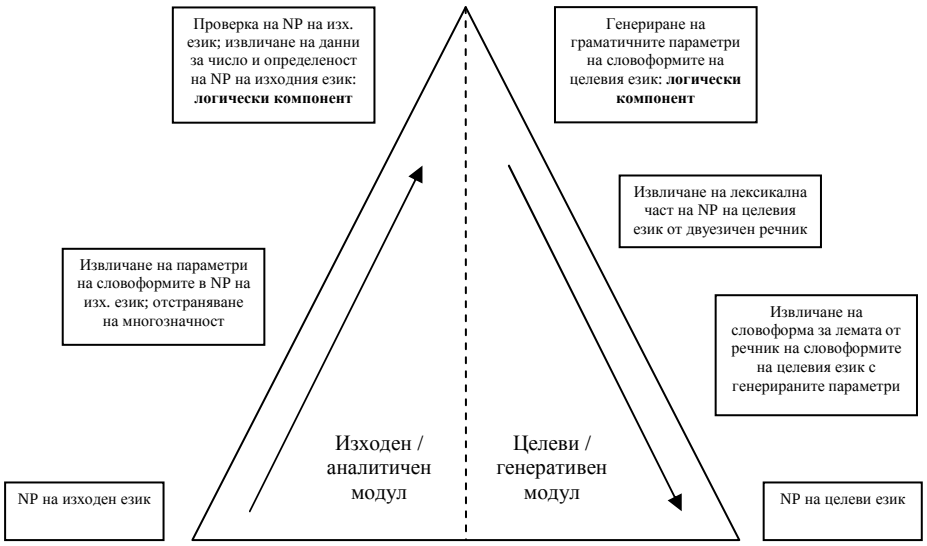
Изводи за функциите

От казаното досега може да се направят следните основни изводи:

- При реализацията на инструмент за машинен превод, който да претендира за възможности за по-широко приложение, не би следвало да се отчитат всички възможни признаци на прилагателното с маркер *–а* на шведски. В логическия компонент на NPTrans той следва да се укаже само като «маркер *–а*»;
- На нивото на трансфер от работата на трансферната система за машинен превод не следва да фигурира признак за род на изходния език;
- На нивото на трансфер от работата на трансферната система за машинен превод следва да фигурира признак за определеност и число на именната фраза на изходния език.

Принципна схема

При спазване на тези изисквания и зададените ограничения може да се състави следната принципна схема за работата на NPTrans:



При наличие на достатъчно богат речник на словоформите на изходния и целевия език и достатъчно богат двуезичен речник структурата на именните фрази, които така описаният инструмент може да анализира и превежда, следва да зависи единствено от правилата, зададени в логическия компонент.

По-долу следва описание на реализацията на инструмента NPTrans по горната принципна схема.

Реализация

Изисквания към системата

Инструментът NPTrans е написан под Windows XP за дистрибуцията *ActivePerl 5.10.0*. За реализацията му са използвани някои библиотеки, които не са част от базовата инсталация на *ActivePerl 5.10.0*. Всички от тях с изключение на `AI::Prolog`[11] могат да се инсталират от графичния интерфейс за управление на пакети *Perl Package Manager*. Те са:

- `XML::Twig`
- `DBI`
- `DBD::mysqlPP`

Функции от изброените библиотеки се използват пряко в кода на NPTrans за анализ и генериране на XML и извличане на данни от базата от данни на MySQL.

Библиотеката `AI::Prolog` се инсталира от архива <http://cpan.uwinnipeg.ca/cpan/authors/id/J/JJ/JJORE/AI-Prolog-0.740.tar.gz> или направо от конзолата `cpan`[14] с командата `install AI::Prolog`. `AI::Prolog` използва функции от следните библиотеки:

- `aliased`
- `Clone`
- `Exporter::Tidy`
- `Hash::AsObject`
- `Hash::Util`
- `Pod::Usage`
- `Regexp::Common`
- `Scalar::Util`
- `Term::ReadKey`
- `Term::ReadLine`
- `Text::Balanced`
- `Text::Quote`

Всички те се инсталират от графичния интерфейс за управление на пакети *Perl Package Manager*. Ако някоя от тези библиотеки липсва, изпълнението на `moduleSource.pl` и `moduleTarget.pl` се прекратява със съобщение за грешка.

[11] Поради спецификата на библиотеката `AI::Prolog` представям реализация на NPTrans за работа на самостоятелен компютър с връзка към мрежата, а не реализация на уебсървър. Това не означава, че такава не се предвижда.

За извличане на данни от речника на словоформите на шведски език в реално време се изисква също така и връзка с интернет. Базата от данни, върху която NPTrans работи, се намира на сървър с IP адрес `88.203.244.110`.

NPTrans изисква и права за четене и писане в папката, в която се инсталира.

Изходен корпус

NPTrans предполага наличието на изходен корпус на шведски език във формат XML с анотация на именните фрази и отделните словоформи, от които те са съставени. Анотирането на изходния корпус може да става отделно от процеса на работа на NPTrans и то не е предмет на настоящата работа. Структурата на корпуса, която се намира в дефиницията `styles/corpus.dtd`, е следната:[12]

```
<!ELEMENT corpus (#PCDATA|NP)*>
<!ELEMENT NP (wform)*>
<!ELEMENT wform (#PCDATA)>
<!ATTLIST NP descr CDATA #IMPLIED>
<!ATTLIST wform global CDATA #IMPLIED lemma CDATA #IMPLIED param CDATA
#IMPLIED word_class CDATA #IMPLIED>
```

За целите на настоящата работа анотирането на изходния корпус се прави от преводача. За удобство може да се ползва редактор за SGML, напр. epcEdit[15].

Примерен вид на изходния текст:

```
All erfarenhet tyder på att statligt ägande har små chanser att
förbättra läget för fordonsbranschen. Snarare finns stor risk att
situationen för Volvo och Saab blir långsiktigt sämre med staten som
ägare. Det framgår av en ny rapport om forskningsläget kring effekterna
av statligt och privat ägande. Hur många bilar som ska tillverkas i
Sverige bör vara en fråga för marknaden, inte för politiken.[13]
```

Примерен вид на изходния корпус:

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
  <NP>
    <wform>All</wform>
    <wform>erfarenhet</wform></NP>tyder på att
  <NP>
    <wform>statligt</wform>
    <wform>ägande</wform></NP>har
  <NP>
    <wform>små</wform>
    <wform>chanser</wform></NP>att förbättra
  <NP>
    <wform>läget</wform></NP>för
  <NP>
    <wform>fordonsbranschen</wform></NP>. Snarare finns
  <NP>
    <wform>stor</wform>
```

[12] Подробно описание на атрибутите на словоформите се съдържа в описанието на аналитичния модул на NPTrans `moduleSource.pl` по-долу.

[13] Статия от шведския ежедневник *Dagens Nyheter* (http://www.dn.se/DNet/jsp/polopoly.jsp?d=572&a=869849). Файлът `corpora/DNArtikel.xml` съдържа пълния текст на статията от 1179 думи в анотиран корпус. За максимална яснота той се представя форматиран с шаблон за трансформация на XML съдържание XSLT (файла `styles/npSource.xsl`). Текстът без анотация се намира във файла `raw/DNArtikel.txt`.

```
    <wform>risk</wform></NP>att
  <NP>
    <wform>situationen</wform></NP>för
  <NP>
    <wform>Volvo</wform>
    <wform>och</wform>
    <wform>Saab</wform></NP>blir långsiktigt sämre med
  <NP>
    <wform>staten</wform></NP>som
  <NP>
    <wform>ägare</wform></NP>.
  <NP>
    <wform>Det</wform></NP>framgår av
  <NP>
    <wform>en</wform>
    <wform>ny</wform>
    <wform>rapport</wform></NP>om
  <NP>
    <wform>forskningsläget</wform></NP>kring
  <NP>
    <wform>effekterna</wform></NP>av
  <NP>
    <wform>statligt</wform>
    <wform>och</wform>
    <wform>privat</wform>
    <wform>ägande</wform></NP>. Hur många
  <NP>
    <wform>bilar</wform></NP>som ska tillverkas i
  <NP>
    <wform>Sverige</wform></NP>bör vara
  <NP>
    <wform>en</wform>
    <wform>fråga</wform></NP>för
  <NP>
    <wform>marknaden</wform></NP>, inte för
  <NP>
    <wform>politiken</wform></NP>.
</corpus>
```

Анотирането на изходния корпус е трудоемка задача, която в едно реално приложение задължително трябва да се автоматизира. За целта може да се използва краен преобразувател за анализ на нивото на изречението след нормализация на изходния текст. Един такъв анализ би могъл да бъде предмет на друга разработка въз основа на опита и изводите от настоящата работа.

Към настоящия момент в NPTrans е предвидено автоматично добавяне на таговете за словоформите в състава на именната фраза `<wform></wform>` посредством модула `wformTagger.pl`. Това става след маркиране на именните фрази в изходния текст от преводача и значително намалява техническата работа. `wformTagger.pl` работи върху изходния корпус на следния етап:

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
  <NP>All erfarenhet</NP>tyder på att
```

```
  <NP>statligt ägande</NP>har
  <NP>små chanser</NP>att förbättra
  <NP>läget</NP>för
  <NP>fordonsbranschen</NP>. Snarare finns
  <NP>stor risk</NP>att
  <NP>situationen</NP>för
  <NP>Volvo</NP> och <NP>Saab</NP>blir långsiktigt sämre med
  <NP>staten</NP>som
  <NP>ägare</NP>.
  <NP>Det</NP>framgår av
  <NP>en ny rapport</NP>om
  <NP>forskningsläget</NP>kring
  <NP>effekterna</NP>av
  <NP>statligt och privat ägande</NP>. Hur många
  <NP>bilar</NP>som ska tillverkas i
  <NP>Sverige</NP>bör vara
  <NP>en fråga</NP>för
  <NP>marknaden</NP>, inte för
  <NP>politiken</NP>.
<corpus>
```

Модулът `wformTagger.pl` изпълнява чисто технически функции. Той използва библиотеката за анализ на XML файлове `XML::Twig`, която предлага тази функционалност, включително и възможност за т. нар. *pretty printing* – форматиране на XML във вид, удобен за прочитане от екрана. След проверка за наличие на изходния файл, зададен като първи аргумент от командния ред, и проверка за възможност за запис на целевия файл, зададен като втори аргумент от командния ред, в `wformTagger.pl` се създава обекта `$annot_xml_twig` от клас `XML::Twig`, в който се анализира изходния файл и при достигане на затварящ таг `</NP>` се задейства подпрограмата `nounPhrase()`. Тя работи върху съдържанието на току-що прочетения таг `<NP></NP>`. То се разделя на словоформи и се записва в масива `@wformArray` с функцията `split()`. След това в хода на изпълнение на цикъл `foreach()` за всеки от елементите на масива се създава нов таг `<wform></wform>` със съдържание съответната словоформа, който е дъщерен за тага `<NP></NP>`. След като приключи анализът на изходния файл, XML съдържанието се записва форматирано във вид, удобен за прочитане от екрана, с параметри `$annot_xml_twig ->set_pretty_print('indented')` и `$annot_xml_twig ->set_pretty_print('record')` в целевия файл.

За файла `raw/detGronaAppletUntagged.xml` описаният процес изглежда така:

1. Изходен файл: `raw/detGronaAppletUntagged.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
<NP>det gröna äpplet</NP></corpus>
```

2. Стартиране на модула `wformTagger.pl` с нужните параметри на командния ред. От основната директория на NPTrans:

```
>perl            wformTagger.pl            raw/detGronaAppletUntagged.xml
raw/detGronaApplet.xml
Wordform tagger. NP Translator by Georgi Iliev 1.00
Analyzing...Done.
Writing to file...Done.
```

3. Целеви файл: `raw/detGronaApplet.xml`

```
<?xml version="1.0" encoding="UTF-8"?>

<corpus>
  <NP>
    <wform>det</wform>
    <wform>gröna</wform>
    <wform>äpplet</wform></NP>
</corpus>
```

С анотирането на именните фрази приключва работата на преводача по подготовката на изходния корпус. Намеса от негова страна се налага още веднъж в работата на NPTrans – за отстраняване на многозначност при извличане на параметрите на словоформите от речника на словоформите на шведски език.

Изходен (аналитичен) модул на NPTrans: `moduleSource.pl`

Представената по-горе принципна схема за работата на NPTrans е разделена на аналитична и генеративна част, които са обособени в два файла изходен код на Perl: `moduleSource.pl` и `moduleTarget.pl`. Те се намират в главната директория на архива `NPTrans.RAR`.

`moduleSource.pl` работи върху изходния корпус на шведски език. За илюстриране на работата му се използва файла `npSource.xml` в директория `raw/`. Той *не* е свързан текст, а поредица от анотирани именни фрази. Файлът `raw/npSource.xml` съдържа всички възможни реализации на именната фраза на шведски език, които NPTrans поддържа към момента, както са описани те в примерите на с. 26 по-горе. Съдържанието на файла `raw/npSource.xml` е показано в Приложение F.

Работата на аналитичния модул `moduleSource.pl` *в резюме*

Файлът `raw/npSource.xml` съдържа анотирани именни фрази без лексикална или граматична информация за тях. Аналитичният компонент `moduleSource.pl` извлича данни за всяка от словоформите в него от таблицата `lexicon` в базата `saldo` и при натъкване на омоними изчаква от потребителя да укаже правилната форма за конкретния случай в диалогов режим. След извличане на параметрите на всички словоформи в състава на дадена именна фраза логическият компонент на `moduleSource.pl`, обособен във външна база данни, написана на Prolog, проверява дали компонентите на именната фраза със съответните параметри могат да се съчетаят граматически правилно в именна фраза на шведски език и ако да, връща сведения за признаците «число» и «определеност» на валидната именна фраза. След това тя се записва текущо в структурата, в която е прочетен изходния XML файл, заедно с всички параметри на словоформите от базата от данни и определените по логически път параметри на цялата именна фраза. Тези параметри се записват в атрибута `descr` на тага `<NP></NP>` за именната фраза и в атрибутите `global`, `lemma`, `param` и `word_class` на таговете `<wform></wform>` за словоформите, от които тя е съставена. При изчерпване на анотираните фрази в изходния файл те се записват в целеви файл със съответните атрибути. Той е крайният резултат от работата на аналитичния модул, т.е. неговият «output», и служи за «input» на генеративния модул `moduleTarget.pl`.

Подробно описание на работата на аналитичния модул **`moduleSource.pl`**

Освен във файла `moduleSource.pl`, изходният код на аналитичния модул се съдържа и във форматиран вид във файла `text/moduleSource.pl.pdf`. В крайния си вид към момента той съдържа 662 реда код на Perl. От ред 5 до ред 9 са обявени библиотеките, които модулът използва. Това са:

- `XML::Twig` за анализ, четене от XML файлове и писане в XML файлове на високо ниво;
- `HTML::Entities` за нормализиране на съдържанието;
- `AI::Prolog` за заявки към база от данни на Prolog;
- `DBI` за връзка с база от данни;
- `Encode` за преобразуване на текст между различни кодови таблици.

Редове от 13 до 32 съдържат кода за проверка за наличие на необходимите файлове за работата на програмата. Това са:

- аргументите от командния ред `$ARGV[0]` и `$ARGV[1]`, указващи съответно изходния файл и целевия файл за работата на програмата. При неправилно указан изходен файл или липса на права за писане в папката на целевия файл програмата дава съобщение за грешка и прекъсва работата си;
- файлът `prolog/db/dbSource.pro`, който съдържа правилата за анализ на именната фраза на шведски език. Ако не бъде намерен, програмата дава съобщение за грешка и прекъсва работата си;
- файлът `raw/nps.txt`, който се инициализира при всяко пускане на програмата. В него на отделен ред се записва във вид, удобен за четене, всяка именна фраза от изходния файл. Това се прави с цел улеснение на потребителя при отстраняване на многозначност, тъй като анализът на XML файла става в поточен режим, т.е. всички елементи на именната фраза се зареждат едва при достигане на затварящ таг `</NP>` (вж. по-долу);
- файлът `log.txt`, към който се препращат всички съобщения за грешка от стандартния изход (екрана) в реално време. От техническа гледна точка това решение не е особено уместно. От друга страна логическият компонент работи с динамично създавани предикати, което води до издаване на предупреждения от

AI::Prolog за липса на дефиниции. Те нямат отношение към работата на аналитичния модул и с цел постигане на по-голяма прегледност са пренасочени към текстов файл.

- файлът prolog/prologTestSE.pro, в който се записват всички динамично генерирани факти на Prolog по време на работата на програмата. Той няма отношение към работата на аналитичния модул, а служи за проверка на логическия компонент.

На ред 44 се прочита базата от данни с правила на Prolog от файла prolog/db/dbSource.pro в масива $prologRules. След това този файл се затваря.

Следва инициализиране на масивите @prologDB, @prologQry и @prologTerms, които се използват за добавяне на факти към базата от данни на Prolog и изпълнение на заявка към нея при прочитане на именна фраза от изходния файл; променливата $keyCount служи за кодиране на конкретните словоформи с единични символи на латиница, тъй като термове на Prolog не може да се задават със символи извън английската азбука. В конкретния случай това са «å», «ä» и «ö» на шведски и кирилицата на български.

На ред 57 се установява връзка с базата от данни на отдалечения сървър. Променливата $dbh сочи към обекта

```
DBI->connect("dbi:mysqlPP:database=saldo;host=88.203.244.110",
"root", "mastermind",{'RaiseError' => 1});
```

и чрез нея се изпълняват заявките за търсене в базата от данни.

Променливата $NP, инициализирана на ред 64, съдържа именната фраза, която се анализира в съответния момент. Тя се прочита от временния файл raw/nps.txt.

На ред 68 и ред 74 се създават два нови обекта от клас XML::Twig ($annot_xml_twig и $print_nounphrases_twig). И двата анализират изходния XML файл. При анализ на изходния файл с $print_nounphrases_twig->parsefile()достигането на затварящ таг </wform> и </NP> е събитие, при което се задействат съответните подпрограми (sub) print_wform_to_file() и print_np_to_file(). Тяхната задача е проста: първата записва съдържанието на току-що прочетения таг <wform></wform> с един интервал в края, а втората добавя символ за край на реда във временния файл raw/nps.txt. Така на всеки ред от него се съдържа по една именна фраза от изходния XML файл.

При анализ на изходния файл с `$annot_xml_twig->parsefile()` (което става след записа на текстовата част на именните фрази във временния файл `raw/nps.txt`) достигането на затварящ таг `</wform>` и `</NP>` е събитие, при което се задействат съответните подпрограми `wform()` и `validateNP()`.

С `wform()` се задават атрибутите `global`, `lemma`, `param` и `word_class` на текущия таг `<wform></wform>`. Те са параметри за словоформата от тага `<wform></wform>`, които се извличат с подпрограмата `wordformSearch()` от таблицата `lexicon` от базата от данни, към която сочи `$dbh`.

`wordformSearch()` изпълнява две основни функции. Първо тя зарежда от таблицата `lexicon` всички записи, за които словоформата (полето `word` на записа) съвпада със съдържанието на текущия таг `<wform></wform>`. Това става със заявката

```
my $sth = $dbh->prepare("SELECT * FROM lexicon where word=\'$wordform\'");
$sth->execute();
```

След това с цикъла `while (my $ref = $sth->fetchrow_arrayref())` всички полета от всички намерени записи от таблицата, за които полето `word` съвпада със съдържанието на текущия таг `<wform></wform>` се преобразуват от `utf8` във вътрешната система за кодиране на символи на Perl, за да могат да бъдат изведени на екрана.

За илюстрация на структурата на записите в `lexicon` следва резултатът от заявката

```
"SELECT * FROM lexicon where word='äpple'"
```

ID	PID	HEAD	POS	WORD	PARAM	INHS
äpple..nn.1	nn_5n_ansikte	äpple	nn	äpple	sg indef nom	n

Полетата «ID» и «PID» съдържат информация за идентификация на записа и семантични параметри, които не са от значение за работата на NPTrans към момента. Инструментът работи с данните от останалите 5 полета:

- «HEAD» – ЛЕМА;
- «POS» – КЛАС НА РЕЧТА;
- «WORD» – СЛОВОФОРМА;
- «PARAM» – ГРАМАТИЧЕСКИ ПАРАМЕТРИ НА СЛОВОФОРМАТА;
- «INHS» – ЛЕКСИКАЛЕН РОД;

Стойностите на тези полета (от 2 до 6) се записват като стойности за двойките от асоциативните масиви (hash), към които сочат елементите на масива @disamb.

След това се прави същинското отстраняване на многозначност с проверка на броя на членовете на масива @disamb, в който са заредени съответните елементи от записите, за които е установено съответствие между словоформите. Ако масивът е празен (т. е. има размер 0), то словоформата не е намерена в базата от данни и за нея автоматично се указва, че е непозната, като на всички атрибути се задава стойност unknown и в масива от факти на Prolog се вмъква факт is_unknown(). Ако масивът съдържа точно един член (т. е. има размер 1), то за словоформата от изходния файл еднозначно са извлечени параметри от базата от данни. Ако масивът съдържа повече от един член (т. е. има размер > 1), в конзолата на последователно номерирани редове се извеждат записите, които съответстват на словоформата, заедно с анализираната текущо именна фраза (стойността на $NP, прочетена като ред от временния файл raw/nps.txt от подпрограмата noun_phrase() при достигане на отварящ таг <NP> при анализ на изходния XML файл с $annot_xml_twig->parse()). Програмата изчаква от потребителя да въведе номера на реда, на който се намира описанието на словоформата за конкретния случай. След това съответните стойности се подават като изходни параметри към подпрограмата convParamsToLocal(), която връща абстрактно еднозначно описание на словоформата $global_descr във вид на символен низ от флагове. Стойностите на полетата от съответния запис «HEAD», «POS» и «PARAM» се записват в локалните променливи $head_descr, $pos_descr и $param_descr и заедно с $global_descr се връщат като резултат от работата на подпрограмата wordformSearch() в подпрограмата wordform(), където се записват в атрибутите на тага <wform></wform> (global, lemma, word_class и param).

Прекият резултат от работата на подпрограмата convParamsToLocal() е абстрактно описание на словоформата във вид на символен низ. Самото то (записано в атрибута global на тага <wform></wform>) би могло да се използва при анализ на нивото на изречението занапред с цел установяване на границите на именната и глаголната фраза по логически способ. На този етап обаче то само фигурира като резултат от работата на аналитичния модул.

Подпрограмата convParamsToLocal() изпълнява една друга важна функция. В нея данните за част на речта, граматичен род, лексикален род, определеност, число, падеж

и степен на прилагателното, предпоставения определителен и неопределителен член, съществителното, местоимението и наречието се преобразуват във факти на Prolog, които се записват в масива `@prologQry`. Тъй като и на шведски, и на български има символи, които не могат да бъдат част от термове на Prolog, за всяка словоформа от състава на именната фраза се задава абстрактно еднобуквено съответствие на латиница с преобразуване на числовата стойност на променливата `$keyCount` в символ с `chr()`. За словоформата «äpple» от примера по-горе това означава, че при приключване на работата на `convParamsToLocal()` масивът `@prologQry` ще съдържа следните факти[14] за абстракцията «d», генерирана като ключ към словоформата «äpple»:

```
is_nn(d). «d (äpple) е съществително.»
is_n(d). «d (äpple) е от род neutrum.»
is_sg(d). «d (äpple) е в единствено число.»
is_indef(d). «d (äpple) е нечленувано.»
is_nom(d). «d (äpple) е в именителен падеж.»
```

При достигане на затварящ таг `</NP>` при анализ на изходния XML файл с `$annot_xml_twig->parse()` се задейства подпрограмата `validateNP()`. Тя работи върху словоформите, които са текстовата част на всички дъщерни елементи (*children*) на текущия таг `<NP></NP>`, фактите на Prolog, зададени за тези словоформи (по-точно за абстрактните ключове към тях) и граматическите правила на Prolog за проверка на именната фраза на шведски, зададени отделно във файла `prolog/db/dbSource.pro` и заредени в масива `$prologRules`.

При проверката на именната фраза в зависимост от броя на елементите в нея (които при зададените ограничения на възможните реализации на именната фраза на шведски, които NPTrans поддържа, не могат да надвишават 5) се проверява дали те могат да се съчетаят в именна фраза от един от четири възможни вида, всеки от които се дефинира като цел на програмата на Prolog: нечленувана в единствено число (`is_si()`), нечленувана в множествено число (`is_pi()`), членувана в единствено число (`is_sd()`) или членувана в множествено число (`is_pd()`).

Това става по следния начин:

- Създава се нов обект – база от данни на Prolog от правилата от масива `$prologRules`: `$prologDB = AI::Prolog->new($prologRules);`
- Установява се броя на членовете на именната фраза и се записва в `$NP_members`;

[14] Приложение C съдържа списък на всички факти, генерирани за словоформи в реално време.

- С цикъла `foreach (@prologQry) {print PROLOG_TEST $_."\.\n"; $prologDB->do("assert($_).")}` в базата от данни `$prologDB` се вмъкват факти за всеки от елементите на именната фраза (`d`, `e`, `f` и т. н.);
- В зависимост от броя на членовете на именната фраза към базата от данни на Prolog се подава заявка за проверка дали именната фраза отговаря на зададените критерии във файла с «граматиката» `prolog/db/dbSource.pro` за нечленувана в единствено число (`is_si()`), нечленувана в множествено число (`is_pi()`), членувана в единствено число (`is_sd()`) или членувана в множествено число (`is_pd()`) именна фраза – т.е. дали съответната цел е удовлетворена.

Ако нито една от целите не бъде удовлетворена, в тага `<NP></NP>` се задава описание `descr="unknown"` и той ще бъде пропуснат от генеративния модул на NPTrans.

При удовлетворение на някоя от целите в тага `<NP></NP>` за именната фраза се задава съответното описание `descr="is_si"`, `descr="is_pi"`, `descr="is_sd"` или `descr="is_pd"`.

При достигане до края на изходния корпус резултатът от работата на аналитичния модул `moduleSource.pl`, се записва в целевия файл, зададен като втори аргумент на командния ред.

За фразата «det gröna äpplet» – «зелената ябълка» описаният процес изглежда така:

1. Изходен файл: `raw/detGronaApplet.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE corpus SYSTEM "../styles/corpus.dtd">
<corpus>
  <NP>
    <wform>det</wform>
    <wform>gröna</wform>
    <wform>äpplet</wform></NP>
</corpus>
```

2. Стартиране на аналитичния модул `moduleSource.pl` с нужните параметри на командния ред. От основната директория на NPTrans: [15]

```
>perl moduleSource.pl raw/detGronaApplet.xml corpora/detGronaApplet.xml
Source Module. NP Translator by Georgi Iliev 1.00
Connecting to SALDO database...Done.
Analyzing...
```

[15] Буквите «ä», «ö» и «å» от шведската азбука не се извеждат правилно на екрана, тъй като Windows използва `cp866` за кирилизация на конзолата. Предпочетох компромис с три от буквите на шведски за сметка на цялата кирилица. Това няма значение за работата на NPTrans по изходния и целевите файлове, тъй като там кодирането е `utf8`.

```
Noun phrase analyzed: det grўna ϕpplet
Searching for "det" in database...
1. LEMMA: den; WORD CLASS: pn; PARAM: sg n nom
2. LEMMA: den; WORD CLASS: al; PARAM: sg n

Disambiguation. Enter number of line containing correct interpretation:
2
Done.
Noun phrase analyzed: det grўna ϕpplet
Searching for "grўna" in database...
1. LEMMA: grўn; WORD CLASS: av; PARAM: pos indef pl nom
2. LEMMA: grўn; WORD CLASS: av; PARAM: pos def sg no_masc nom
3. LEMMA: grўn; WORD CLASS: av; PARAM: pos def pl nom

Disambiguation. Enter number of line containing correct interpretation:
2
Done.
Noun phrase analyzed: det grўna ϕpplet
Searching for "ϕpplet" in database...
Done.

Validating...Done.
A total of 3 wordforms in source file.
3 wordforms found in database and annotated.
```

3. Факти за членовете на именната фраза (записани в реално време във временен файл за проверка `prolog/prologTestSE.pro`):

```
is_al(d).
is_av(e).
is_def(e).
is_def(f).
is_n(d).
is_n(f).
is_nn(f).
is_no_masc(e).
is_nom(e).
is_nom(f).
is_pos(e).
is_sg(d).
is_sg(e).
is_sg(f).
```

4. Съдържание на изходния файл `corpora/detGronaApplet.xml` при приключване на работата на аналитичния модул `moduleSource.pl`:

```
<?xml version="1.0" encoding="UTF-8"?>
<corpus>
  <NP descr="is_sd">
    <wform global="as----n---" lemma="den" param="sg n"
word_class="al">det</wform>
    <wform global="2sxNp0----" lemma="grön" param="pos def sg no_masc
nom" word_class="av">gröna</wform>
    <wform global="1sxN---n--" lemma="äpple" param="sg def nom"
word_class="nn">äpplet</wform>
  </NP>
</corpus>
```

Целеви (генеративен) модул на NPTrans: `moduleTarget.pl`

Работата на генеративния модул `moduleTarget.pl` *в резюме*

Генеративният модул `moduleTarget.pl` работи върху продукта от работата на аналитичния модул – XML файл, съдържащ анотирани именни фрази на изходния език, където всеки таг за словоформа `<wform></wform>` съдържа информация за съответната словоформа като стойности на атрибутите `global`, `lemma`, `param` и `word_class`, а всеки таг за именна фраза `<NP></NP>` съдържа информация за числото и членуването на фразата като стойност на атрибута `descr`.

Генеративният модул прочита последователно така анотираните именни фрази, извличайки лексикалната част на всяка именна фраза. Това се налага поради наличието в шведския на предпоставен неопределителен и определителен член, какъвто на български няма. След това за всеки лексикален компонент на съответната именна фраза генеративният модул прочита съответната лема и извлича информация за съответната част на речта. Освен това извлича и информация за числото и членуването на съответната фраза. Въз основа на тази информация логическият компонент на генеративния модул `moduleTarget.pl`, обособен във външна база от данни от правила на Prolog, генерира граматическите параметри за всеки от лексикалните членове на именната фраза на български език.

В отделен текстов файл е обособен двуезичен речник. В него се търси лемата на български, която съответства на лемата на шведски, и при успех от речника на словоформите на български във формат DELAF се извлича конкретната словоформа по генерираните граматически параметри. Същата се добавя текущо към символен низ, който представлява съответната именна фраза на български. Ако в двуезичния речник не бъде намерено съответствие на лемата на шведски, генеративният модул пристъпва към работа по следващата именна фраза от изходния файл.

Подробно описание на работата на генеративния модул `moduleTarget.pl`

Освен във файла `moduleTarget.pl`, изходният код на генеративния модул се съдържа и във форматиран вид във файла `text/moduleTarget.pl.pdf`. В крайния си вид към момента той съдържа 568 реда код на Perl. От ред 5 до ред 8 са обявени библиотеките, които модулът използва. Това са:

- `XML::Twig` за анализ, четене от XML файлове и писане в XML файлове на високо ниво;
- `HTML::Entities` за нормализиране на съдържанието;
- `AI::Prolog` за заявки към база от данни на Prolog;
- `Encode` за преобразуване на текст между различни кодови таблици.

Редове 12 до 41 съдържат кода за проверка за наличие на необходимите файлове за работата на програмата. Това са:

- аргументите от командния ред `$ARGV[0]` и `$ARGV[1]`, указващи съответно изходния файл и целевия файл за работата на програмата. При неправилно указан изходен файл или липса на права за писане в папката на целевия файл програмата дава съобщение за грешка и прекъсва работата си.
- файлът `prolog/db/dbTarget.pro`, който съдържа правилата за генериране на параметрите на словоформите в състава на именната фраза на български език. Ако не бъде намерен, програмата дава съобщение за грешка и прекъсва работата си;
- файлът `dic/sweBul.dic`, който съдържа двуезичен шведско-български речник. Ако не бъде намерен, програмата дава съобщение за грешка и прекъсва работата си;
- файлът `delaf/bul.dic`, който съдържа речник на словоформите на български език във формат DELAF. Ако не бъде намерен, програмата дава съобщение за грешка и прекъсва работата си;
- файлът `log.txt`, към който се препращат всички съобщения за грешка от стандартния изход (екрана) в реално време. От техническа гледна точка това решение не е особено уместно. От друга страна логическият компонент работи с динамично създавани предикати, което води до издаване на предупреждения от `AI::Prolog` за липса на дефиниции. Те нямат отношение към работата на генеративния модул и с цел постигане на по-голяма прегледност са пренасочени към текстов файл;
- файлът `prolog/prologTestBG.pro`, в който се записват всички динамично генерирани факти на Prolog по време на работата на програмата. Той няма

отношение към работата на генеративния модул, а служи за проверка на логическия компонент.

На ред 50 се прочита базата от данни от правила на Prolog от файла `prolog/db/dbTarget.pro` в масива `$prologRules`. След това този файл се затваря.

Следва инициализиране на масивите `@prologDB` и `@prologQry`, които се използват за добавяне на факти към базата от данни на Prolog и изпълнение на заявка към нея при генериране на параметрите на отделните членове на именната фраза на целевия език.

На ред 64 цялото съдържание на файла `delaf/bul.dic` се прочита от подпрограмата `delafRead()` в структура от паметта, към която сочи променливата `$refLoadedDelaf`. Файлът `delaf/bul.dic` се чете поредово. Пример за реда, на който се намира словоформата «ябълката»:

```
ябълката,ябълка.N+F:sd
```

Така прочетеният ред се разделя на словоформа `wordform` (от началото на реда до запетаята), компонент за лема `lemma` (от запетаята до точката), компонент за лексикални параметри `semParams` (от точката до първото двоеточие) и компонент за граматични параметри `gramParams` (от първото двоеточие до края на реда). От своя страна компонентът за граматични параметри се подразделя на съответния брой елементи в зависимост от това колко различни интерпретации са възможни за конкретната словоформа.

В резултат от това съдържанието на файла `delaf/bul.dic` се зарежда в масив от референтни променливи, всяка от които сочи към асоциативен масив, съставен от четири двойки ключ-стойност (ключове `wordform` `lemma` `semParams` със съответни стойности словоформа, лема и лексикални параметри от реда и ключ `gramParams`, чиято стойност е референтна променлива, която сочи към масив от символни низове, в който се записват всички интерпретации на граматичните параметри за конкретната словоформа), с индекс номер на съответния ред от файла `delaf/bul.dic`.

За пример можем да разгледаме ред 7 от файла `delaf/bul.dic`:

```
австрийски,австрийски.A:s:p
```

Словоформата `австрийски` е прилагателно (`A`) с лема `австрийски` и съответства на форма за единствено число `s` или форма за множествено число `p`. След прочитане на

файла delaf/bul.dic в паметта с подпрограмата delafRead() в структурата, към която сочи променливата $refLoadedDelaf, на ред 7 съответства следното:

```
$VAR7 = {
          'gramParams' => [
                            's',
                            'p'
                          ],
          'lemma' => 'австрийски',
          'wordform' => 'австрийски',
          'semParams' => 'A'
        };
```

След прочитане в паметта файла delaf/bul.dic се затваря.

На ред 76 цялото съдържание на двуезичния речник от файла dic/sweBul.dic се прочита от подпрограмата readBilingual() в масива @bilingual. Файлът dic/sweBul.dic се чете поредово. Пример за реда, на който се намира словоформата «äpple»:

```
äpple ябълка
```

Така прочетеният ред се разделя на компонент за изходен език swe (от началото на реда до интервала) и компонент за целеви език bul (от интервала до края на реда). В резултат от това съдържанието на файла dic/sweBul.dic се зарежда в масив от референтни променливи, всяка от които сочи към асоциативен масив от две двойки ключ-стойност (ключ swe със стойност лемата на изходния език и ключ bul със стойност лемата на целевия език), с индекс номера на реда от речника. След прочитане на файла dic/sweBul.dic в паметта с подпрограмата readBilingual() в масива @bilingual, на ред 3 от речника съответства следната структура в паметта:

```
$VAR3 = {
          'bul' => 'ябълка',
          'swe' => 'äpple'
        };
```

След прочитане в паметта файла dic/sweBul.dic се затваря.

На ред 82 се създава нов обект от клас XML::Twig ($annot_xml_twig).

При анализ на изходния файл с $annot_xml_twig->parsefile() достигането на затварящ таг </NP> е събитие, при което се задейства подпрограмата extractNPSemantics().

extractNPSemantics() работи върху цялото съдържание на прочетения таг <NP></NP>. Първоначално се прочита описанието на именната фраза (стойността на атрибута descr

на тага <NP></NP>) и от всички дъщерни елементи на тага <NP></NP> (т.е. таговете <wform></wform>) се отделят лексикалните елементи, които ще служат за основа при генерирането на именната фраза на целевия език. Това са всички елементи от изходния език с изключение на предпоставения неопределителен или определителен член.

Ако аналитичният модул не е разпознал конкретната именна фраза (стойност на атрибута descr на тага <NP></NP> "unknown"), генеративният модул moduleTarget.pl издава съобщение за грешка и преминава към следващата именна фаза от изходния файл. Аналогична е процедурата и при наличие на непознати словоформи.

Ако именната фраза е била разпозната от аналитичния модул като един от четирите вида (is_si, is_pi, is_sd или is_pd) и за всички словоформи е налице еднозначно описание[16], за всички лексикални компоненти на именната фраза на изходния език подред се прави търсене в двуезичния речник (масива @bilingual). Ако някой от тях не бъде намерен в речника, генеративният модул moduleTarget.pl издава съобщение за грешка и преминава към следващата именна фраза от изходния файл.

Ако всички лексикални компоненти на именната фраза на изходния език фигурират в двуезичния речник, за съответствието на целевия език на всяка от лемите за тях подред се прави търсене в речника на словоформите DELAF с подпрограмата extractSemanticsFromDelaf(), която извлича от речника на словоформите, към който сочи $refLoadedDelaf, информация за лексикалния род на думата. Тя се съдържа в символния низ, който представлява стойност за ключа 'semParams' от съответния запис в $refLoadedDelaf. За «ябълки» този запис изглежда така:

```
$VAR = {
          'gramParams' => [
                            'p'
                          ],
          'lemma' => 'ябълка',
          'wordform' => 'ябълки',
          'semParams' => 'N+F'
        };
```

а лексикалният род на думата е указан като (+)F.

Всички параметри на словоформите в състава на именната фраза на целевия език се генерират въз основа на описанието на именната фраза на изходния език,

[16] В случая се прави двойна проверка, тъй като аналитичният модул няма как да определи характеристиките на именната фраза без да има данни за параметрите на всички словоформи в нея.

информацията за частите на речта, от които е съставена, и рода на съществителното-опора на целевия език.

При установяване на лексикалния род за съответната еднобуквена абстракция, генерирана за означаване на конкретната част от именната фраза на целевия език, в масива от факти на Prolog `@prologQry` се добавя съответния предикат.[17] Поради специфичната организация на речника на словоформите SALDO и наличието на падежна форма на съществителното на шведски за притежание (родителен падеж с окончание –s) се налага изрично добавяне на предикати, указващи родителен падеж и притежателна форма на местоимението.[18]

Описанието на именната фраза на изходния език заедно с асоциативния масив, в който лексикалните компоненти на именната фраза на целевия език са кодирани с еднобуквени ключове, се подават като аргументи към подпрограмата `npFormTranslate()`. Нейната задача е въз основа на направено запитване към базата от данни с правила на Prolog от външната «граматика» (файла `prolog/db/dbTarget.pro`, прочетен в `$prologRules`) и вмъкнатите впоследствие факти за абстракциите, които съответстват на лексикалните компоненти на именната фраза на целевия език (лексикален род, падеж и притежателна форма), да генерира необходимите символни низове, които описват точно необходимата словоформа от речника на словоформите на български език във формат DELAF.

`npFormTranslate()` връща като резултат от работата си масив, в който са подредени последователно символните низове, по които се прави търсене в граматическите признаци от речника на словоформите. За вече анотираната именна фраза «det gröna äpplet» – «зелената ябълка» във файла, който получихме в резултат от работата на аналитичния модул по-горе `corpora/detGronaApplet.xml`, този процес протича така:

1. Изходен файл: `corpora/detGronaApplet.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
```

[17] Вж. Приложение С

[18] В речника на словоформите SALDO притежателните местоимения са записани под лема съответното лично местоимение. Пример: лема за «min» е «jag». Това представлява проблем за работата на NPTrans, тъй като в речника на словоформите на български във формат DELAF, лемата на притежателното местоимение е различна от лемата на личното местоимение. Пример: лема за «мои» е «мой». С цел илюстриране на работата на NPTrans към настоящия момент е направен компромис с този факт и за вариантите на именната фраза с притежателно местоимение в двуезичния речник за лема – лично местоимение на изходния език е зададена лема – притежателно местоимение на целевия език. Пример: «`jag` `мой`». Проблеми като този навеждат на мисълта, че се налага и въвеждане на семантичен план в двуезичния речник.

```
<corpus>
  <NP descr="is_sd">
    <wform global="as----n---" lemma="den" param="sg n"
word_class="al">det</wform>
    <wform global="2sxNp0----" lemma="grön" param="pos def sg no_masc
nom" word_class="av">gröna</wform>
    <wform global="1sxN---n--" lemma="äpple" param="sg def nom"
word_class="nn">äpplet</wform>
  </NP>
</corpus>
```

2. Стартиране на генеративния модул `moduleTarget.pl` с нужните параметри на командния ред. От основната директория на NPTrans:

```
>perl moduleTarget.pl corpora/detGronaApplet.xml detGronaApplet.txt
Target Module. NP Translator by Georgi Iliev 1.00
NOW READING DELAF...Done.
DELAF LOADED: SIZE 12091
Analyzing NP "det grўna фpplet"...
Searching for "зелен" in DELAF database...found.
Searching for "ябълка" in DELAF database...found.
prologDescription is_sd
paramString: sfd
paramString: s
Extracting wordform for "зелен" from DELAF database...Done.
Extracting wordform for "ябълка" from DELAF database...Done.
```

3. Факти за членовете на именната фраза (записани в реално време във временен файл за проверка `prolog/prologTestBG.pro`):

```
is_av(d).
is_nn(e).
is_sd(d,e).
is_sem_f(e).
```

4. Съдържание на изходния файл `detGronaApplet.txt` при приключване на работата на генеративния модул `moduleTarget.pl`:

```
det gröna äpplet => зелената ябълка
```

Проверка на заявената функционалност на NPTrans

Функционалност на аналитичния модул на NPTrans: `moduleSource.pl`

За проверка на заявената функционалност служи файлът `raw/npSource.xml`, който съдържа всички модели на именни фрази на шведски език, които е предвидено NPTrans да анализира и разпознава на изходния език, а оттам и да превежда на целевия език.

Команден ред (от основната директория на NPTrans):

```
>perl moduleSource.pl raw/npSource.xml corpora/npSource.xml
```

Резултатът от работата на аналитичния модул (файла `corpora/npSource.xml`) се съдържа в Приложение G. За максимална яснота той се представя форматиран с шаблон за трансформация на XML съдържание XSLT (файла `styles/npSource.xsl`).

Функционалност на генеративния модул на NPTrans: `moduleTarget.pl`

За проверка на заявената функционалност служи файлът `corpora/npSource.xml`, генериран като резултат от работата на аналитичния модул.

Команден ред (от основната директория на NPTrans):

```
>perl moduleTarget.pl corpora/npSource.xml corpora/npTarget.txt
```

Изходът в конзолата от работата на генеративния модул се съдържа в Приложение H.

В резултат от работата на генеративния модул се създава текстов файл `corpora/npTarget.txt` със следното съдържание:

```
bil => кола
bilar => коли
bilen => колата
bilarna => колите
en bil => кола
fint väder => хубаво време
en fin blomma => хубаво цвете
fina blommor => хубави цветя
en blomma => едно цвете
en vacker blomma => едно красиво цвете
en mycket vacker blomma => едно много красиво цвете
tre blommor => три цветя
tre vackra blommor => три красиви цветя
tre mycket vackra blommor => три много красиви цветя
de tre bilarna => трите коли
de tre svarta bilarna => трите черни коли
de tre alldeles svarta bilarna => трите напълно черни коли
svarta tavlan => черната дъска
det höga huset => високата къща
de höga husen => високите къщи
```

en mycket god macka => много вкусен сандвич
mycket goda mackor => много вкусни сандвичи
den mycket goda mackan => много вкусния сандвич
de mycket goda mackorna => много вкусните сандвичи
hennes klänning => нейната рокля
hennes fina klänning => нейната хубава рокля
hans skor => неговите обувки
hans bruna skor => неговите кафяви обувки

Изводи

Поради ограниченото време за работа и липсата на достатъчно ресурси в настоящия си вид NPTrans няма практическо приложение. Анотирането на именните фрази на изходния език и отстраняването на многозначност става ръчно, което прави работата с инструмента твърде тромава.

Двуезичният речник е съставен единствено за целите на превод на използваните примери за илюстриране на работата на NPTrans. Академичното преподаване на скандинавски езици у нас започва преди по-малко от 15 години и съвсем естествено първите речници и помагала за съответните езици се появяват в този период. Няма широко достъпни двуезични речници между шведски и български, които да могат да се използват от инструмент от вида на NPTrans, представен с настоящата работа. В отсъствието на богат двуезичен речник NPTrans може да се преобразува в система за управление на двуезична база от данни, която текущо се разширява от преводач. Освен това, форматът на двуезичния речник е твърде опростен. Тъй като записите в базата от данни SALDO съдържат и семантична информация, в по-нататъшната работа по NPTrans може да се въведе семантичен компонент, който да служи за връзка между двуезичния речник на лемите и речника на словоформите на изходния език.

Първоначалното намерение при пристъпване към работа по настоящия проблем бе да се използват непроменени вече съществуващи езикови ресурси (речника на словоформите SALDO на шведски и този във формат DELAF на български език). Но при въвеждане на функционалност за превод на фрази с притежателни местоимения се оказа, че за анотиране на притежателните местоимения на шведски е използвана различна система от тази в речника на български език (вж. бележка под линия 18 на с. 50 по-горе). Това наложи въвеждането на промяна в начина, по който притежателните местоимения са систематизирани в речника на словоформите на български. Тази промяна представлява сериозен компромис с първоначално заложените принципи, поради което в работата занапред следва да се предвиди автоматично привеждане на формите на местоименията (а вероятно и на други части на речта) на български език към вид, който от една страна позволява ефективна работа с NPTrans, а от друга – не налага промени във формата на изходните бази от данни.

Използваната принципна схема с обособен логически компонент даде много добри резултати, що се отнася до възможностите за усъвършенстване на граматиката и

разширяване на обхвата от синтактични конструкции, които NPTrans поддържа. Това дава основание за оптимизъм за бъдещето приложение ако не на самия инструмент NPTrans, то поне на зададения с него модел на система за машинен превод, базиран на правила. Затова се предвижда следващата стъпка от работата по NPTrans да бъде конфигуриране за работа на уебсървър с възможност за отдалечен достъп и задаване на граматически правила по удобен за потребителя начин. За момента това не бе възможно, тъй като работата на логическия компонент зависи от твърде много други библиотеки, всяка от които трябва да присъства на отдалечения сървър.[19] Но резултатите от направените опити на отдалечен сървър са окуражаващи.

За предоставяне на NPTrans като инструмент за машинен превод за потребителски достъп по мрежата е необходимо:

- автоматизирано установяване на границите на именната фраза в текста. Това може да стане с прилагане на логически компонент за анализ на еднозначно установените параметри на отделни словоформи, разположението на неизменяемите части на речта, а конкретно за шведски и с отчитане на строгия словоред на подлога, сказуемото и централното обстоятелствено пояснение (*satsadverbial*);
- автоматизирано отстраняване на многозначност на нивото на именната фраза на изходния език. Въз основа на разработените логически модели за проверка на именната фраза на изходния език досега е възможно да се прави проверка за граматично съответствие на именни фрази, съставени от всички възможни интерпретации на конкретните словоформи, получени като резултат от търсенето в базата от данни. Така може с висока степен на сигурност да се посочва най-вероятната интерпретация, при условие, разбира се, че изходният текст е граматически правилен;
- оптимизиране на логическия компонент – базата от данни на Prolog `prolog/db/dbSource.pro` за изходния език и `prolog/db/dbTarget.pro` за целевия език и отстраняване на многократните проверки за едни и същи характеристики с цел постигане на по-висока степен на универсалност.

[19] Вж. с. 31

Използвани източници

[1] Arnold, D.J., L. Balkan, S. Meijer, R. Lee. Humphreys and L. Sadler. Machine Translation: an Introductory Guide. London: NCC Blackwell, 1994, 173-195

[2] Tufiş, D., Sv. Koeva, T. Erjavec, M. Gavrilidou, C. Krstev. «Building Language Resources and Translation Models for Machine Translation focused on South Slavic and Balkan Languages.» In: Tadić et al. (eds). Proceedings of The Sixth International Conference Formal Approaches to South Slavic and Balkan Languages, 25-28 September 2008. Dubrovnik, pp. 145-153, 2008

[3] Коева, Св. «Съвременни езикови технологии – приложения и перспективи.» В: Закони на/за езика. София: Хейзъл, 2004, 111-157

[4] Red. Holm, Britta och Nylund, Elizabeth. Deskriptiv svensk grammatik. Stockholm: Liber AB, 1970

[5] Wessén, Elias. Vårt svenska språk. Stockholm: A&W, 1968

[6] Ницолова, Р. Българска граматика. Морфология. София: УИ, 2008

[7] *Saldos hemsida.* 01 Dec 2008. Lars Borin, Markus Forsberg och Lennart Lönngren. 01 Dec 2008 – 15 Jan 2009 <http://spraakbanken.gu.se/sal/>

[8] *The Generalized Example-Based Machine Translation project*. 29 Nov 2004. Language Technologies Institute (LTI) at Carnegie Mellon University (CMU). 10 Jan – 15 Jan 2009 <http://www.cs.cmu.edu/~ralf/ebmt.html>

[9] *SYSTRAN Homepage*. 2008. SYSTRAN. 10 Jan – 15 Jan 2009 <http://www.systran.co.uk/>

[10] *Google Translate Homepage*. 2009. Google. 10 Jan – 15 Jan 2009 <http://translate.google.com>

[11] *Strawberry Prolog Homepage.* 2009. Dimiter Dimitrov Dobrev. 15 Dec 2008 – 15 Jan 2009 <http://www.dobrev.com/>

[12] *TEI: Text Encoding Initiative Homepage*. The Text Encoding Initiative Consortium. 10 Jan – 15 Jan 2009 <http://www.tei-c.org/>

[13] *EuroMatrix Project website.* Euromatrix. 10 Jan – 15 Jan 2009 <http://www.euromatrix.net/>

[14] *Comprehensive Perl Archive Network.* CPAN. 15 Dec 2008 – 15 Jan 2009 <http://www.cpan.org/>

[15] *epcEdit XML/SGML editor*. Heinz Detlev Koch & Roman Halstenberg. 15 Dec 2008 – 15 Jan 2009 <http://www.epcedit.com/>

[16] *The JRC-Acquis Multilingual Parallel Corpus Homepage*. European Commission Joint Research Centre. 10 Jan – 15 Jan 2009 <http://langtech.jrc.it/JRC-Acquis.html>

За автора

*Авторът е бакалавър по английска филология от СУ (2002 г.) и бакалавър по скандинавистика от СУ (2009 г.) с основен език шведски и втори език датски и по професия е преводач с 8-годишен опит в превода на техническа, правна, научна и търговска литература от/на английски език и скандинавски езици. От две години насам работи по обществена поръчка за превод на общоевропейска база от данни за търговски марки, регистрирани в ЕС, която се превежда от всички езици на ЕС на български с помощта на система за машинен превод (*ESTeam Translator*,* <http://www.esteam.se/>*). С* ESTeam Translator *работи по превода от шведски и датски език. В областта на компютърната лингвистика интересите на автора са съсредоточени главно върху проблемите на машинния превод.*

За връзка: georgi.iliev@roboread.com

Приложение А: Означения, използвани за описание на словоформите на шведски в базата от данни *SALDO*[20]

Части на речта (колона «POS» от записа)	
al	предпоставен определителен или неопределителен член
av	прилагателно име
ab	наречие
nl	числително име
nn	съществително име
pn	местоимение

Граматически параметри на словоформата (колона «PARAM» от записа)	
sg	единствено число
pl	множествено число
n	род *neutrum*
u	род *utrum*
nom	именителна форма
poss	притежателна форма на местоимението
def	членувано
indef	нечленувано
pos	положителна форма на прилагателното
num	числително бройно
ord	числително редно
masc	форма на прилагателното/числителното редно само за м. р.
no_masc	форма на прилагателното/числителното редно без маркер за м. р.

Лексикални параметри на лемата (колона «INHS» от записа)	
u	род *utrum*
n	род *neutrum*

[20] Посочват се само онези, които се поддържат от NPTrans

Приложение В: Означения, използвани за описание на словоформите на български в базата от данни *DELAF*[21]

Граматически параметри на словоформата (символен низ след първото двоеточие на реда)	
s	единствено число
p	множествено число
z	множествено число – числително
m	мъжки род
f	женски род
n	среден род
d	членувано

Лексикални параметри на словоформата (символен низ между точката и първото двоеточие на реда)	
M	мъжки род
F	женски род
N	среден род

[21] Посочват се само онези, които се поддържат от NPTrans

Приложение C: Факти на Prolog, описващи параметрите на словоформите

На шведски

is_al(wform).	«wform е предпоставен неопределителен/определителен член.»
is_av(wform).	«wform е прилагателно име.»
is_ab(wform).	«wform е наречие.»
is_nl(wform).	«wform е числително име.»
is_nn(wform).	«wform е съществително име.»
is_pn(wform).	«wform е местоимение.»
is_sg(wform).	«wform е в единствено число.»
is_pl(wform).	«wform е в множествено число.»
is_n(wform).	«wform е от род *neutrum*.»
is_u(wform).	«wform е от род *utrum*.»
is_poss(wform).	«wform е притежателна форма на местоимение.»
is_def(wform).	«wform е определена/членувана форма.»
is_indef(wform).	«wform е неопределена/нечленувана форма.»
is_a_marked(wform).	«wform е форма на прилагателно с маркер *-a*.»

На български

is_av(wform).	«wform е прилагателно име.»
is_ab(wform).	«wform е наречие.»
is_nl(wform).	«wform е числително име.»
is_nn(wform).	«wform е съществително име.»
is_pn(wform).	«wform е местоимение.»
is_sg(wform).	«wform е в единствено число.»
is_pl(wform).	«wform е в множествено число.»
is_m(wform).	«wform е от мъжки род.»
is_f(wform).	«wform е от женски род.»
is_n(wform).	«wform е от среден род.»
is_sem_m(wform).	«wform е (съществително) от мъжки род.»
is_sem_f(wform).	«wform е (съществително) от женски род.»
is_sem_n(wform).	«wform е (съществително) от среден род.»
is_poss(wform).	«wform е притежателна форма на местоимение.»
is_def(wform).	«wform е определена/членувана форма.»
is_indef(wform).	«wform е неопределена/нечленувана форма.»

Приложение D: Правила за проверка на именната фраза на шведски (`prolog/db/dbSource.pro`)

```
agrees(X,Y) :- is_al(X), is_sg(X), is_n(X), is_def(Y), is_sg(Y),
is_n(Y).
agrees(X,Y) :- is_al(X), is_sg(X), is_u(X), is_def(Y), is_sg(Y),
is_u(Y).
agrees(X,Y) :- is_al(X), is_sg(X), is_n(X), is_indef(Y), is_sg(Y),
is_n(Y).
agrees(X,Y) :- is_al(X), is_sg(X), is_u(X), is_indef(Y), is_sg(Y),
is_u(Y).
agrees(X,Y) :- is_al(X), is_pl(X), is_def(Y), is_pl(Y).
agrees(X,Y) :- is_def(X), is_sg(X), is_n(X), is_def(Y), is_sg(Y),
is_n(Y).
agrees(X,Y) :- is_def(X), is_sg(X), is_u(X), is_def(Y), is_sg(Y),
is_u(Y).
agrees(X,Y) :- is_def(X), is_pl(X), is_def(Y), is_pl(Y).
agrees(X,Y) :- is_indef(X), is_sg(X), is_n(X), is_indef(Y), is_sg(Y),
is_n(Y).
agrees(X,Y) :- is_indef(X), is_sg(X), is_u(X), is_indef(Y), is_sg(Y),
is_u(Y).
agrees(X,Y) :- is_indef(X), is_pl(X), is_indef(Y), is_pl(Y).
agrees(X,Y) :- is_av(X), is_a_marked(X), is_pl(Y).
agrees(X,Y) :- is_av(X), is_a_marked(X), is_def(Y).
agrees(X,Y) :- is_nl(X), is_num(X), is_nn(Y), is_u(X), is_u(Y),
is_sg(X), is_sg(Y).
agrees(X,Y) :- is_nl(X), is_num(X), is_nn(Y), is_u(X), is_u(Y),
is_pl(X), is_pl(Y).
agrees(X,Y) :- is_nl(X), is_num(X), is_nn(Y), is_n(X), is_n(Y),
is_sg(X), is_sg(Y).
agrees(X,Y) :- is_nl(X), is_num(X), is_nn(Y), is_n(X), is_n(Y),
is_pl(X), is_pl(Y).
agrees(X,Y) :- is_pn(X), is_poss(X), is_nn(Y), is_u(X), is_u(Y),
is_sg(X), is_sg(Y),  is_indef(Y).
agrees(X,Y) :- is_pn(X), is_poss(X), is_nn(Y), is_n(X), is_n(Y),
is_sg(X), is_sg(Y), is_indef(Y).
agrees(X,Y) :- is_pn(X), is_poss(X), is_nn(Y), is_pl(X), is_pl(Y),
is_indef(Y).

is_a_marked(X) :- is_av(X), is_pl(X).
is_a_marked(X) :- is_av(X), is_def(X).

is_pi(X) :- is_nn(X), is_indef(X), is_pl(X).
is_pi(X,Y) :- is_av(X), is_a_marked(X), is_nn(Y), agrees(X,Y).
is_pi(X,Y) :- is_nl(X), is_pl(X), is_nn(Y), is_indef(Y), is_pl(Y).
is_pi(A,X,Y) :- is_ab(A), is_av(X), is_a_marked(X), is_nn(Y), is_pl(Y),
is_indef(Y).
is_pi(A,X,Y) :- is_nl(A), is_av(X), agrees(A,Y), is_nn(Y), is_pl(Y),
is_indef(Y), agrees(X,Y).
is_pi(A,B,X,Y) :- is_nl(A), is_ab(B), agrees(A,Y), is_av(X),
is_a_marked(X), is_nn(Y), is_pl(Y), is_indef(Y).

is_pd(X) :- is_nn(X), is_def(X), is_pl(X).
is_pd(A,X) :- is_pn(A), is_poss(A), agrees(A,X), is_nn(X), is_pl(X).
is_pd(X,Y,Z) :- is_al(X), is_pl(X), is_av(Y), is_a_marked(Y), is_nn(Z),
agrees(X,Z), is_def(Z).
is_pd(A,X,Y) :- is_pn(A), is_poss(A), agrees(A,Y), is_av(X),
is_a_marked(X), is_nn(Y), is_pl(Y).
is_pd(A,B,X) :- is_al(A), is_pl(A), is_nl(B), is_pl(B), is_nn(X),
agrees(A,X), is_def(X).
```

```
is_pd(X,A,Y,Z) :- is_al(X), is_pl(X), is_ab(A), is_av(Y),
is_a_marked(Y), is_nn(Z), agrees(X,Z), is_def(Z).
is_pd(A,B,X,Y) :- is_al(A), is_pl(A), is_nl(B), is_pl(B), is_av(X),
is_a_marked(X), is_nn(Y), agrees(A,Y), is_def(Y).
is_pd(A,B,C,X,Y) :- is_al(A), is_pl(A), is_nl(B), is_pl(B), is_ab(C),
is_av(X), is_a_marked(X), is_nn(Y), agrees(A,Y), is_def(Y).

is_sd(X) :- is_nn(X), is_def(X), is_sg(X).
is_sd(X,Y) :- is_av(X), is_a_marked(X), is_nn(Y), is_sg(Y), is_def(Y).
is_sd(A,X) :- is_pn(A), is_poss(A), agrees(A,X), is_nn(X), is_sg(X).
is_sd(X,Y,Z) :- is_al(X), is_sg(X), is_av(Y), is_a_marked(Y), is_nn(Z),
agrees(X,Z).
is_sd(A,X,Y) :- is_pn(A), is_poss(A), agrees(A,Y), is_av(X),
is_a_marked(X), is_nn(Y), is_sg(Y).
is_sd(X,A,Y,Z) :- is_al(X), is_sg(X), is_ab(A), is_av(Y),
is_a_marked(Y), is_nn(Z), agrees(X,Z), is_def(Z).

is_si(X) :- is_nn(X), is_indef(X), is_sg(X).
is_si(X,A,Y,Z) :- is_al(X), is_sg(X), is_ab(A), is_av(Y), is_nn(Z),
agrees(X,Z), agrees(Y,Z).
is_si(A,B,X,Y) :- is_nl(A), is_sg(A), is_ab(B), is_av(X), agrees(A,Y),
is_nn(Y), is_sg(Y), is_indef(Y), agrees(X,Y).
is_si(X,Y) :- is_al(X), is_sg(X), is_nn(Y), agrees(X,Y).
is_si(X,Y) :- is_av(X), is_nn(Y), agrees(X,Y), is_sg(Y).
is_si(X,Y) :- is_nl(X), is_sg(X), is_nn(Y), is_indef(Y), is_sg(Y),
agrees(X,Y).
is_si(X,Y,Z) :- is_al(X), is_sg(X), is_av(Y), agrees(X,Z), is_nn(Z),
agrees(Y,Z).
is_si(A,X,Y) :- is_nl(A), is_av(X), agrees(A,Y), is_nn(Y), is_sg(Y),
is_indef(Y), agrees(X,Y).
```

Приложение E: Правила за генериране на именната фраза на български (`prolog/db/dbTarget.pro`)

```
is_invar(A) :- is_sd(A,X,Y), is_ab(A).
is_invar(A) :- is_pd(A,X,Y), is_ab(A).
is_invar(A) :- is_si(A,X,Y), is_ab(A).
is_invar(A) :- is_pi(A,X,Y), is_ab(A).
is_invar(B) :- is_si(A,B,X,Y), is_nl(A), is_ab(B).
is_invar(B) :- is_pi(A,B,X,Y), is_nl(A), is_ab(B).
is_invar(B) :- is_pd(A,B,X,Y), is_nl(A), is_ab(B).

is_def(X) :- is_sd(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_def(X) :- is_pd(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_def(A) :- is_sd(A,X,Y), is_pn(A), is_poss(A), is_av(X), is_nn(Y).
is_def(A) :- is_pd(A,X,Y), is_pn(A), is_poss(A), is_av(X), is_nn(Y).
is_def(X) :- is_sd(X,Y), is_av(X), is_nn(Y).
is_def(X) :- is_pd(X,Y), is_av(X), is_nn(Y).
is_def(A) :- is_sd(A,X), is_pn(A), is_poss(A), is_nn(X).
is_def(A) :- is_pd(A,X), is_pn(A), is_poss(A), is_nn(X).
is_def(X) :- is_sd(X).
is_def(X) :- is_pd(X).
is_def(A) :- is_pd(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
is_def(A) :- is_pd(A,X), is_nl(A), is_nn(X).
is_def(A) :- is_pd(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_nn(Y).

is_indef(Y) :- is_sd(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_indef(Y) :- is_pd(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_indef(Y) :- is_sd(X,Y), is_av(X), is_nn(Y).
is_indef(Y) :- is_pd(X,Y), is_av(X), is_nn(Y).
is_indef(Y) :- is_pd(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_nn(Y).
is_indef(X) :- is_sd(A,X), is_pn(A), is_poss(A), is_nn(X).
is_indef(X) :- is_sd(A,X,Y), is_pn(A), is_poss(A), is_av(X), is_nn(Y).
is_indef(X) :- is_pd(A,X,Y), is_pn(A), is_poss(A), is_av(X), is_nn(Y).
is_indef(X) :- is_si(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_indef(Y) :- is_si(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_indef(X) :- is_si(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
is_indef(Y) :- is_si(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
is_indef(X) :- is_si(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
is_indef(X) :- is_pi(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_indef(Y) :- is_pi(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
is_indef(X) :- is_si(X,Y), is_av(X), is_nn(Y).
is_indef(Y) :- is_si(X,Y), is_av(X), is_nn(Y).
is_indef(X) :- is_pi(X,Y), is_av(X), is_nn(Y).
is_indef(X) :- is_pi(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
is_indef(Y) :- is_pi(X,Y), is_av(X), is_nn(Y).
is_indef(X) :- is_si(X).
is_indef(X) :- is_pi(X).
is_indef(X) :- is_pd(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_nn(Y).

agrees(X,Y) :- is_sd(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
agrees(X,Y) :- is_si(A,X,Y), is_ab(A), is_av(X), is_nn(Y).
agrees(X,Y) :- is_sd(X,Y), is_av(X), is_nn(Y).
agrees(X,Y) :- is_si(X,Y), is_av(X), is_nn(Y).
agrees(X,Y) :- is_si(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
agrees(X,Y) :- is_si(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_nn(Y).

is_f(X) :- agrees(X,Y), is_sem_f(Y), is_sg(Y).
is_f(A) :- is_si(A,X,Y), is_nl(A), is_av(X), is_sem_f(Y).
is_f(A) :- is_si(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_sem_f(Y).
is_f(X) :- is_sd(A,X,Y), is_pn(A), is_poss(A), is_av(X), is_sem_f(Y).
```

```
is_f(A) :- is_sd(A,X,Y), is_pn(A), is_poss(A), is_av(X), is_sem_f(Y).
is_f(A) :- is_sd(A,X), is_pn(A), is_poss(A), is_nn(X), is_sem_f(X).
is_f(A) :- is_si(A,X), is_nl(A), is_nn(X), is_sem_f(X).

is_m(X) :- agrees(X,Y), is_sem_m(Y), is_sg(Y).
is_m(A) :- is_si(A,X,Y), is_nl(A), is_av(X), is_sem_m(Y).
is_m(A) :- is_si(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_sem_m(Y).
is_m(X) :- is_sd(A,X,Y), is_poss(A), is_av(X), is_sem_m(Y).
is_m(A) :- is_sd(A,X), is_pn(A), is_poss(A), is_nn(X), is_sem_m(X).
is_m(A) :- is_si(A,X), is_nl(A), is_nn(X), is_sem_m(X).

is_n(X) :- agrees(X,Y), is_sem_n(Y), is_sg(Y).
is_n(A) :- is_si(A,X,Y), is_nl(A), is_av(X), is_sem_n(Y).
is_n(A) :- is_si(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_sem_n(Y).
is_n(X) :- is_sd(A,X,Y), is_poss(A), is_av(X), is_sem_n(Y).
is_n(A) :- is_sd(A,X), is_pn(A), is_poss(A), is_nn(X), is_sem_n(X).
is_n(A) :- is_si(A,X), is_nl(A), is_nn(X), is_sem_n(X).

is_z(A) :- is_pi(A,X), is_nl(A), is_nn(X).
is_z(A) :- is_pi(A,X,Y), is_nl(A), is_av(X), is_nn(Y).
is_z(A) :- is_pi(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_nn(Y).
is_z(A) :- is_pd(A,X,Y), is_nl(A), is_pl(A), is_av(X), is_nn(Y).
is_z(A) :- is_pd(A,B,X,Y), is_nl(A), is_ab(B), is_av(X), is_nn(Y).
is_z(A) :- is_pd(A,X), is_nl(A), is_nn(X).

is_pl(X) :- is_pd(A,X,Y).
is_pl(X) :- is_pi(A,X,Y).
is_pl(Y) :- is_pd(A,X,Y).
is_pl(Y) :- is_pi(A,X,Y).
is_pl(X) :- is_pi(A,B,X,Y).
is_pl(Y) :- is_pi(A,B,X,Y).
is_pl(Y) :- is_pd(A,B,X,Y).
is_pl(X) :- is_pd(A,B,X,Y).
is_pl(X) :- is_pd(X,Y).
is_pl(X) :- is_pi(X,Y).
is_pl(Y) :- is_pd(X,Y).
is_pl(Y) :- is_pi(X,Y).
is_pl(X) :- is_pd(X).
is_pl(X) :- is_pi(X).
is_pl(A) :- is_pd(A,X,Y).
is_pl(A) :- is_pd(A,X).

is_sg(X) :- is_sd(A,X,Y).
is_sg(Y) :- is_sd(A,X,Y).
is_sg(X) :- is_si(A,X,Y).
is_sg(Y) :- is_si(A,X,Y).
is_sg(A) :- is_sd(A,X,Y).
is_sg(A) :- is_sd(A,X).
is_sg(X) :- is_si(A,B,X,Y).
is_sg(Y) :- is_si(A,B,X,Y).
is_sg(X) :- is_sd(X,Y).
is_sg(Y) :- is_sd(X,Y).
is_sg(X) :- is_si(X,Y).
is_sg(Y) :- is_si(X,Y).
is_sg(X) :- is_sd(X).
is_sg(X) :- is_si(X).
is_sg(A) :- is_si(A,X,Y).
is_sg(A) :- is_si(A,B,X,Y).
```

Приложение F: Съдържание на файла `raw/npSource.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE corpus SYSTEM "../styles/corpus.dtd">
<corpus>
  <NP>
    <wform>bil</wform></NP>
  <NP>
    <wform>bilar</wform></NP>
  <NP>
    <wform>bilen</wform></NP>
  <NP>
    <wform>bilarna</wform></NP>
  <NP>
    <wform>en</wform><wform>bil</wform></NP>
  <NP>
    <wform>fint</wform><wform>väder</wform></NP>
  <NP>
    <wform>en</wform><wform>fin</wform><wform>blomma</wform></NP>
  <NP>
    <wform>fina</wform><wform>blommor</wform></NP>
  <NP>
    <wform>en</wform><wform>blomma</wform></NP>
  <NP>
    <wform>en</wform><wform>vacker</wform><wform>blomma</wform></NP>
  <NP>
    <wform>en</wform><wform>mycket</wform><wform>vacker</wform>
    <wform>blomma</wform></NP>
  <NP>
    <wform>tre</wform><wform>blommor</wform></NP>
  <NP>
    <wform>tre</wform><wform>vackra</wform><wform>blommor</wform></NP>
  <NP>
    <wform>tre</wform><wform>mycket</wform><wform>vackra</wform>
    <wform>blommor</wform></NP>
  <NP>
    <wform>de</wform><wform>två</wform><wform>bilarna</wform></NP>
  <NP>
    <wform>de</wform><wform>två</wform><wform>svarta</wform>
    <wform>bilarna</wform></NP>
  <NP>
    <wform>de</wform><wform>två</wform><wform>alldeles</wform>
    <wform>svarta</wform><wform>bilarna</wform></NP>
  <NP>
    <wform>svarta</wform><wform>tavlan</wform></NP>
  <NP>
    <wform>det</wform><wform>höga</wform><wform>huset</wform></NP>
  <NP>
    <wform>de</wform><wform>höga</wform><wform>husen</wform></NP>
  <NP>
    <wform>en</wform><wform>mycket</wform><wform>god</wform>
    <wform>macka</wform></NP>
  <NP>
    <wform>mycket</wform><wform>goda</wform><wform>mackor</wform></NP>
  <NP>
    <wform>den</wform><wform>mycket</wform><wform>goda</wform>
    <wform>mackan</wform></NP>
  <NP>
    <wform>de</wform><wform>mycket</wform><wform>goda</wform>
    <wform>mackorna</wform></NP>
  <NP>
```

```
    <wform>hennes</wform><wform>klänning</wform></NP>
  <NP>
    <wform>hennes</wform><wform>fina</wform><wform>klänning</wform></NP>
  <NP>
    <wform>hans</wform><wform>skor</wform></NP>
  <NP>
    <wform>hans</wform><wform>bruna</wform><wform>skor</wform></NP>
</corpus>
```

Приложение G: Съдържание на файла `corpora/npSource.xml`

NP descr: is_si

bil

nn

`sg indef nom`

NP descr: is_pi

bilar

nn

`pl indef nom`

NP descr: is_sd

bilen

nn

`sg def nom`

NP descr: is_pd

bilarna

nn

`pl def nom`

NP descr: is_si

en	*bil*
al	**nn**
`sg u`	`sg indef nom`

NP descr: is_si

fint	*väder*
av	**nn**
`pos indef sg n nom`	`sg indef nom`

NP descr: is_si

en	*fin*	*blomma*
al	**av**	**nn**
`sg u`	`pos indef sg u nom`	`sg indef nom`

NP descr: is_pi

fina	*blommor*

av	nn
pos indef pl nom	pl indef nom

NP descr: is_si

en	*blomma*
nl	**nn**
nom num u	sg indef nom

NP descr: is_si

en	*vacker*	*blomma*
nl	**av**	**nn**
nom num u	pos indef sg u nom	sg indef nom

NP descr: is_si

en	*mycket*	*vacker*	*blomma*
nl	**ab**	**av**	**nn**
nom num u	pos	pos indef sg u nom	sg indef nom

NP descr: is_pi

tre	*blommor*
nl	**nn**
nom num u	pl indef nom

NP descr: is_pi

tre	*vackra*	*blommor*
nl	**av**	**nn**
nom num u	pos indef pl nom	pl indef nom

NP descr: is_pi

tre	*mycket*	*vackra*	*blommor*
nl	**ab**	**av**	**nn**
nom num u	pos	pos indef pl nom	pl indef nom

NP descr: is_pd

de	tre	*bilarna*
al	**nl**	**nn**
pl	nom num u	pl def nom

NP descr: is_pd

de	tre	*svarta*	*bilarna*
al	**nl**	**av**	**nn**
pl	nom num u	pos def pl nom	pl def nom

NP descr: is_pd

de	tre	*alldeles*	*svarta*	*bilarna*
al	**nl**	**ab**	**av**	**nn**
pl	nom num u	invar	pos def pl nom	pl def nom

NP descr: is_sd

svarta	*tavlan*
av	**nn**
pos def sg no_masc nom	sg def nom

NP descr: is_sd

det	*höga*	*huset*
al	**av**	**nn**
sg n	pos def sg no_masc nom	sg def nom

NP descr: is_pd

de	*höga*	*husen*
al	**av**	**nn**
pl	pos def pl nom	pl def nom

NP descr: is_si

en	*mycket*	*god*	*macka*
al	**ab**	**av**	**nn**
sg u	pos	pos indef sg u nom	sg indef nom

NP descr: is_pi

mycket	*goda*	*mackor*
ab	**av**	**nn**
pos	pos indef pl nom	pl indef nom

NP descr: is_sd

den	*mycket*	*goda*	*mackan*
al	**ab**	**av**	**nn**
sg u	pos	pos def sg no_masc nom	sg def nom

NP descr: is_pd

de	*mycket*	*goda*	*mackorna*
al	**ab**	**av**	**nn**
pl	pos	pos def pl nom	pl def nom

NP descr: is_sd

hennes	*klänning*
pn	**nn**
poss sg u	sg indef nom

NP descr: is_sd

hennes	*fina*	*klänning*
pn	**av**	**nn**
poss sg u	pos def sg no_masc nom	sg indef nom

NP descr: is_pd

hans	*skor*
pn	**nn**
poss pl	pl indef nom

NP descr: is_pd

hans	*bruna*	*skor*
pn	**av**	**nn**
poss pl	pos def pl nom	pl indef nom

Приложение Н: Изход в конзолата по време на работа на генеративния модул

```
>perl moduleTarget.pl corpora/npSource.xml corpora/npTarget.txt
Target Module. NP Translator by Georgi Iliev 1.00
NOW READING DELAF...Done.
DELAF LOADED: SIZE 12105
Analyzing NP "bil"...
Searching for "кола" in DELAF database...found.
prologDescription is_si
paramString: s
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "bilar"...
Searching for "кола" in DELAF database...found.
prologDescription is_pi
paramString: p
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "bilen"...
Searching for "кола" in DELAF database...found.
prologDescription is_sd
paramString: sd
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "bilarna"...
Searching for "кола" in DELAF database...found.
prologDescription is_pd
paramString: pd
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "en bil"...
Searching for "кола" in DELAF database...found.
prologDescription is_si
paramString: s
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "fint vϕder"...
Searching for "хубав" in DELAF database...found.
Searching for "време" in DELAF database...found.
prologDescription is_si
paramString: sn
paramString: s
Extracting wordform for "хубав" from DELAF database...Done.
Extracting wordform for "време" from DELAF database...Done.
Analyzing NP "en fin blomma"...
Searching for "хубав" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_si
paramString: sn
paramString: s
Extracting wordform for "хубав" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "fina blommor"...
Searching for "хубав" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_pi
paramString: p
paramString: p
Extracting wordform for "хубав" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "en blomma"...
Searching for "един" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_si
```

```
paramString: sn
paramString: s
Extracting wordform for "един" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "en vacker blomma"...
Searching for "един" in DELAF database...found.
Searching for "красив" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_si
paramString: sn
paramString: sn
paramString: s
Extracting wordform for "един" from DELAF database...Done.
Extracting wordform for "красив" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "en mycket vacker blomma"...
Searching for "един" in DELAF database...found.
Searching for "много" in DELAF database...found.
Searching for "красив" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_si
paramString: sn
paramString: INVAR
paramString: sn
paramString: s
Extracting wordform for "един" from DELAF database...Done.
Extracting wordform for "много" from DELAF database...Done.
Extracting wordform for "красив" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "tre blommor"...
Searching for "три" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_pi
paramString: z
paramString: p
Extracting wordform for "три" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "tre vackra blommor"...
Searching for "три" in DELAF database...found.
Searching for "красив" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_pi
paramString: z
paramString: p
paramString: p
Extracting wordform for "три" from DELAF database...Done.
Extracting wordform for "красив" from DELAF database...Done.
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "tre mycket vackra blommor"...
Searching for "три" in DELAF database...found.
Searching for "много" in DELAF database...found.
Searching for "красив" in DELAF database...found.
Searching for "цвете" in DELAF database...found.
prologDescription is_pi
paramString: z
paramString: INVAR
paramString: p
paramString: p
Extracting wordform for "три" from DELAF database...Done.
Extracting wordform for "много" from DELAF database...Done.
Extracting wordform for "красив" from DELAF database...Done.
```

```
Extracting wordform for "цвете" from DELAF database...Done.
Analyzing NP "de tre bilarna"...
Searching for "три" in DELAF database...found.
Searching for "кола" in DELAF database...found.
prologDescription is_pd
paramString: zd
paramString: p
Extracting wordform for "три" from DELAF database...Done.
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "de tre svarta bilarna"...
Searching for "три" in DELAF database...found.
Searching for "черен" in DELAF database...found.
Searching for "кола" in DELAF database...found.
prologDescription is_pd
paramString: zd
paramString: p
paramString: p
Extracting wordform for "три" from DELAF database...Done.
Extracting wordform for "черен" from DELAF database...Done.
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "de tre alldeles svarta bilarna"...
Searching for "три" in DELAF database...found.
Searching for "напълно" in DELAF database...found.
Searching for "черен" in DELAF database...found.
Searching for "кола" in DELAF database...found.
prologDescription is_pd
paramString: zd
paramString: INVAR
paramString: p
paramString: p
Extracting wordform for "три" from DELAF database...Done.
Extracting wordform for "напълно" from DELAF database...Done.
Extracting wordform for "черен" from DELAF database...Done.
Extracting wordform for "кола" from DELAF database...Done.
Analyzing NP "svarta tavlan"...
Searching for "черен" in DELAF database...found.
Searching for "дъска" in DELAF database...found.
prologDescription is_sd
paramString: sfd
paramString: s
Extracting wordform for "черен" from DELAF database...Done.
Extracting wordform for "дъска" from DELAF database...Done.
Analyzing NP "det hŭga huset"...
Searching for "висок" in DELAF database...found.
Searching for "къща" in DELAF database...found.
prologDescription is_sd
paramString: sfd
paramString: s
Extracting wordform for "висок" from DELAF database...Done.
Extracting wordform for "къща" from DELAF database...Done.
Analyzing NP "de hŭga husen"...
Searching for "висок" in DELAF database...found.
Searching for "къща" in DELAF database...found.
prologDescription is_pd
paramString: pd
paramString: p
Extracting wordform for "висок" from DELAF database...Done.
Extracting wordform for "къща" from DELAF database...Done.
Analyzing NP "en mycket god macka"...
Searching for "много" in DELAF database...found.
Searching for "вкусен" in DELAF database...found.
```

```
Searching for "сандвич" in DELAF database...found.
prologDescription is_si
paramString: INVAR
paramString: sm
paramString: s
Extracting wordform for "много" from DELAF database...Done.
Extracting wordform for "вкусен" from DELAF database...Done.
Extracting wordform for "сандвич" from DELAF database...Done.
Analyzing NP "mycket goda mackor"...
Searching for "много" in DELAF database...found.
Searching for "вкусен" in DELAF database...found.
Searching for "сандвич" in DELAF database...found.
prologDescription is_pi
paramString: INVAR
paramString: p
paramString: p
Extracting wordform for "много" from DELAF database...Done.
Extracting wordform for "вкусен" from DELAF database...Done.
Extracting wordform for "сандвич" from DELAF database...Done.
Analyzing NP "den mycket goda mackan"...
Searching for "много" in DELAF database...found.
Searching for "вкусен" in DELAF database...found.
Searching for "сандвич" in DELAF database...found.
prologDescription is_sd
paramString: INVAR
paramString: smd
paramString: s
Extracting wordform for "много" from DELAF database...Done.
Extracting wordform for "вкусен" from DELAF database...Done.
Extracting wordform for "сандвич" from DELAF database...Done.
Analyzing NP "de mycket goda mackorna"...
Searching for "много" in DELAF database...found.
Searching for "вкусен" in DELAF database...found.
Searching for "сандвич" in DELAF database...found.
prologDescription is_pd
paramString: INVAR
paramString: pd
paramString: p
Extracting wordform for "много" from DELAF database...Done.
Extracting wordform for "вкусен" from DELAF database...Done.
Extracting wordform for "сандвич" from DELAF database...Done.
Analyzing NP "hennes klɸnning"...
Searching for "неин" in DELAF database...found.
Searching for "рокля" in DELAF database...found.
prologDescription is_sd
paramString: sfd
paramString: s
Extracting wordform for "неин" from DELAF database...Done.
Extracting wordform for "рокля" from DELAF database...Done.
Analyzing NP "hennes fina klɸnning"...
Searching for "неин" in DELAF database...found.
Searching for "хубав" in DELAF database...found.
Searching for "рокля" in DELAF database...found.
prologDescription is_sd
paramString: sfd
paramString: sf
paramString: s
Extracting wordform for "неин" from DELAF database...Done.
Extracting wordform for "хубав" from DELAF database...Done.
Extracting wordform for "рокля" from DELAF database...Done.
Analyzing NP "hans skor"...
```

```
Searching for "негов" in DELAF database...found.
Searching for "обувка" in DELAF database...found.
prologDescription is_pd
paramString: pd
paramString: p
Extracting wordform for "негов" from DELAF database...Done.
Extracting wordform for "обувка" from DELAF database...Done.
Analyzing NP "hans bruna skor"...
Searching for "негов" in DELAF database...found.
Searching for "кафяв" in DELAF database...found.
Searching for "обувка" in DELAF database...found.
prologDescription is_pd
paramString: pd
paramString: p
paramString: p
Extracting wordform for "негов" from DELAF database...Done.
Extracting wordform for "кафяв" from DELAF database...Done.
Extracting wordform for "обувка" from DELAF database...Done.
```

Приложение I: Списък с файловете в архива `NPTrans.rar`

```
moduleSource.pl
moduleTarget.pl
wformTagger.pl
corpora/detGronaApplet.xml
corpora/DNArtikel.xml
corpora/npSource.xml
delaf/bul.dic
dic/sweBul.dic
prolog/db/dbSource.pro
prolog/db/dbTarget.pro
raw/detGronaApplet.xml
raw/detGronaAppletUntagged.xml
raw/DNArtikel.txt
raw/DNArtikel.xml
raw/npSource.xml
styles/corpus.dtd
styles/npSource.xsl
text/moduleSource.pl.pdf
text/moduleTarget.pl.pdf
```