

Implementierung einer Lateingrammatik im Grammatical Framework

Kolloquium Computerlinguistisches Arbeiten SS 2013

Herbert Lange

10. Juni 2013

Inhalt

1 Einführung

- Das Grammatical Framework
- Die Ressource Grammar Library
- Die Lateinische Sprache

2 Umsetzung

- Lexikon
- Morphologie
- Syntax
- Ausblick

3 Literatur

Das Grammatical Framework

- Mächtigkeit äquivalent zu PMCFG (Parallel Multiple Context-Free Grammars) → zwischen mild und voll kontextsensitiv
- Trennung von abstrakter und konkreter Syntax
- Verschiedene API-Ebenen und Einbindungsmöglichkeiten (u.a. Java, JavaScript)

Die Ressource Grammar Library

- Minimaler Satz gemeinsamer Bestandteile verschiedener Sprachen (Beispielvokabular, Wort-/Satzarten, Syntaxregeln)
- ca. 38 Sprachen voll oder teilweise umgesetzt
- ca. 43 geschlossene Kategorien (Determiner, ...) und Phrasentypen
- ca. 22 offene Kategorien (Nomen, Verben, Adjektive, ...)

```
Lang> p ''der Mann sieht die Frau''
```

```
PhrUtt NoPConj (UttS (UseCl (TTAnt TPres ASimul) PPos (PredVP (DetCN
(DetQuant DefArt NumSg) (UseN man_N)) (ComplSlash (SlashV2a see_V2)
(DetCN (DetQuant DefArt NumSg) (UseN woman_N)))))) NoVoc
```

Funktion	Typ	Argumente	Übersetzung
PhrUtt	Phr	(PConj) Utt (Voc)	der Mann sieht die Frau
NoPConj	PConj		(Keine Konjunktion)
UttS	Utt	S	der Mann sieht die Frau
UseCl	S	(Temp) (Pol) Cl	der Mann sieht die Frau
TTAnt	Temp	Tense Ant	(Tempus und Aspekt)
TPres	Tense		(Präsens)
ASimul	Ant		(Gleichzeitigkeit)
PPos	Pol		(Positive Aussage)
PredVP	Cl	NP VP	der Mann sehen die Frau
DetCN	NP	Det CN	der Mann
DetQuant	Det	Quant Num	der
DefArt	Quant		der
NumSg	Num		(Singular)
UseN	CN	N	Mann
man_N	N		Mann
ComplSlash	VP	VPSlash NP	sehen die Frau
		...	
NoVoc	Voc		(Keine Anrede)

Die Lateinische Sprache

Teil der indogermanischen Sprachfamilie → Ähnlichkeiten zu germanischen Sprachen so wie dem Griechischen

lateinisch	griechisch	deutsch
pater	patér	Vater
ager	agrós	Acker
trēs	treĩs	drei
decem	déka	zehn
est	estí	ist

Ursprünglich Sprache der Bewohner der mittelitalienischen Region Latium

Sprachliche Besonderheiten

- Sehr freie Satzstellung → aber meist Verwendung von Subjekt-Objekt-Verb
- Flektierende Sprache mit synthetischer Syntax (Abl. abs. und Acl)
 - Augusto regente pax erat in toto imperio romano → Als/weil Augustus regierte, herrschte im ganzen römischen Reich Frieden.
 - Imperatorem venire audit → Er hört, dass der Imperator kommt.
- Ablativ und Vokativ als eigene Fälle

Umsetzung

- Lexikon der RGL
- Morphologie der RGL
- Syntax der RGL
- Optional: rein sprachspezifische Konstrukte

Lexikon

- Ca. 400 vorgegebene englische Wörter aus allen möglichen Wortarten und verschiedenen Bereichen
- Keine Klärung von Ambiguitäten (z.B. bank) → willkürliche Wahl der Übersetzung
- Viele moderne Begriffe (Auto, Eisenbahn, Computer, Fernseher) → Übersetzung mit Hilfe der Wikipedia
- Begriffe ohne auffindbare Übersetzung (ein-/ausschalten) → Wahl eines nahe liegenden Wortes (accendere/exstinguere)
- Manche Differenzierungen nicht in jeder Sprache möglich

Beispiel

```
blue_A = mkA ( variants 'caeruleus' ; 'caerulus' ) ; -- 3 L...
boat_N = mkN 'navicula' ; -- -ae f. L...
book_N = mkN 'liber' 'libri' masculine ; -- Ranta; -bri m. L...
boot_N = mkN 'calceus' ; -- -i m. L...
boss_N = mkN 'dux' 'ducis' ( variants feminine ; masculine ) ; --
ducis m./f. L...
boy_N = mkN 'puer' 'pueri' masculine ; -- -eri m. L...
bread_N = ( variants (mkN 'panis' 'panis' masculine ) ; mkN
'pane' ) ; -- -is m./n. L...
break_V2 = mkV2 (mkV 'rumpo' 'rupi' 'ruptum' 'rumpere') ; --
Ranta; 3 L...
```

Morphologie

- Einteilung in vier Gruppen: Nomina (Substantive, Adjektive, Pronomina, Numeralia), Verben, Partikel (Adverbien, Präpositionen, Konjunktionen), Interjektionen. Es gibt keine Artikel
- Nomina und Verben werden flektiert, Partikel und Interjektionen werden nicht flektiert
- Fünf Nomendeklinationen, drei Adjektivdeklinationen, Adjektivkomparation, Adverbbildung, vier Verbkonjugationen, Deponentia

Smart Paradigms

Pattern Matching um für ein Wort aus möglichst wenig Formen das Paradigma zu bestimmen

Beispiel

```
noun : Str -> Noun = \verbum ->
case verbum of {
_ + 'a' => noun1 verbum ;
_ + 'us' => noun2us verbum ;
_ + 'um' => noun2um verbum ;
_ + 'er' => noun2er verbum ;
_ + 'u' => noun4u verbum ;
_ + 'es' => noun5 verbum ;
_ => noun3 verbum
} ;
```

Syntax

- Problem: Freie Wortstellung im Satz → Variationen erhöhen die Komplexität exponentiell
- Zunächst: Beschränkung auf SOV-Wortstellung (Subjekt-Objekt-Verb)
- Später: Suche nach performanten Implementierungsmöglichkeiten für beschränkte oder volle Variation

Ausblick

- Anbinden eines größeren Lexikons
- Integration in eine Weboberfläche oder eine Android-App
- Anwendung in der Lehre (Translation Quiz)

Zusammenfassung

- Lexikon mit ca. 400 teils modernen Wörtern
- Morphologie mit Smart Paradigms
- Syntax mit möglichst freier Wortstellung

Literatur

- Bayer, Karl u. Lindauer, Josef: Lateinische Grammatik, C.C. Buchners Verlag, Bamberg 1994
- Leuman, M./Hofmann, J.B./ Szantyr, A.: Lateinische Grammatik. Auf der Grundlage des Werkes von Friedrich Stolz und Joseph Hermann Schmalz; Band 2: Lateinische Syntax und Stilistik, C.H. Beck'sche Verlagsbuchhandlung, München 1972
- Ranta, Aarne: Grammatical Framework. Programming with Multilingual Grammars, CSLI Publications, Stanford 2011
- Ranta, Aarne: Grammatical Framework Tutorial, <http://www.grammaticalframework.org/doc/tutorial/gf-tutorial.html> 2010