The GF Ecosystem

Aarne Ranta
aarne@{chalmers.se,digitalgrammars.com}

Sixth GF Summer School
Stellenbosch 3-14 December 2018



digital Grammars
Language technology to rely on.

The mission of GF

Formalize the grammars of the world

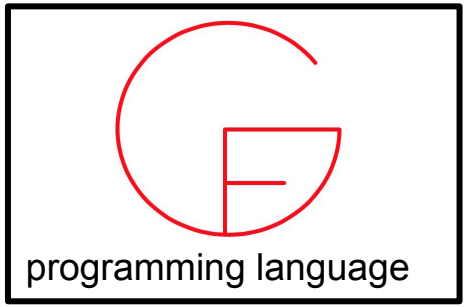
and make them available for computer applications

The challenge

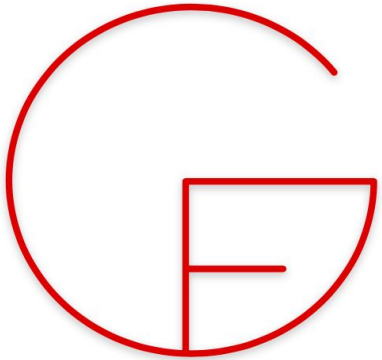
grammars
PhD in linguistics
5 years of work
limited coverage
brittle
obsolete

The challenge

statistics	**grammars**
BSc in computer science	PhD in linguistics
5 weeks of waiting	5 years of work
wide coverage	limited coverage
robust	brittle
state of the art	obsolete


programming language

http://www.grammaticalframework.org/



Grammatical Framework

A programming language for multilingual grammar applications

Get started

- Google Tech Talk
- GF Cloud
- Tutorial

Download GF

Learn more

- The GF Book
- Reference Manual
- Shell Reference
- Best Practices [PDF]

RGL Synopsis

Develop

- Developers Guide
- PGF library API (Haskell runtime)
- PGF library API (C runtime)
- GF compiler API
- Text Editor Support

Contribute

- Mailing List
- Issue Tracker
- Authors
- Summer School

GF on GitHub

GF, a functional programming language

Inspirations:

- ALF
- Haskell, ML
- XFST
- compilers

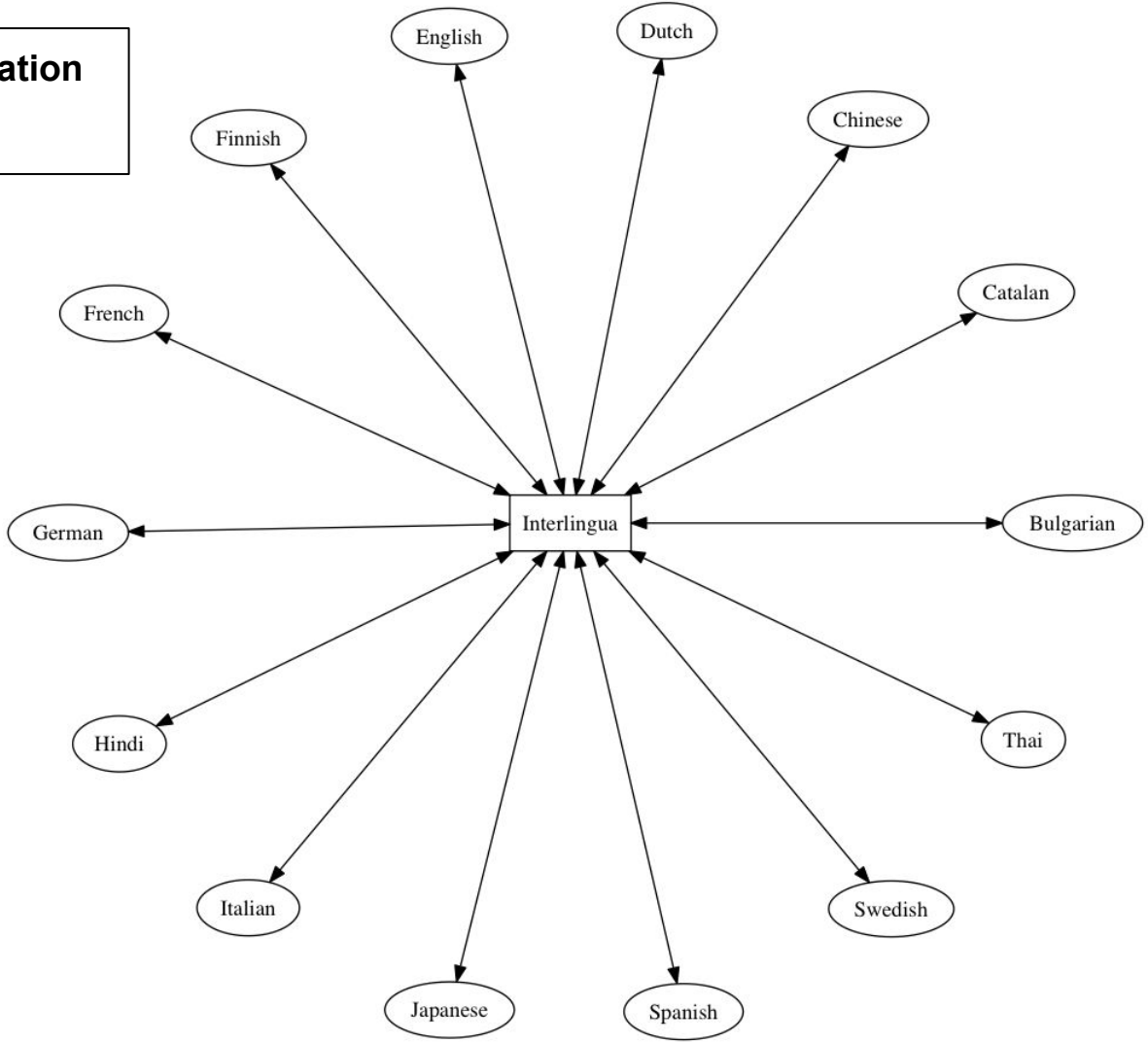
Features

- multilingual
- reversible
- modular

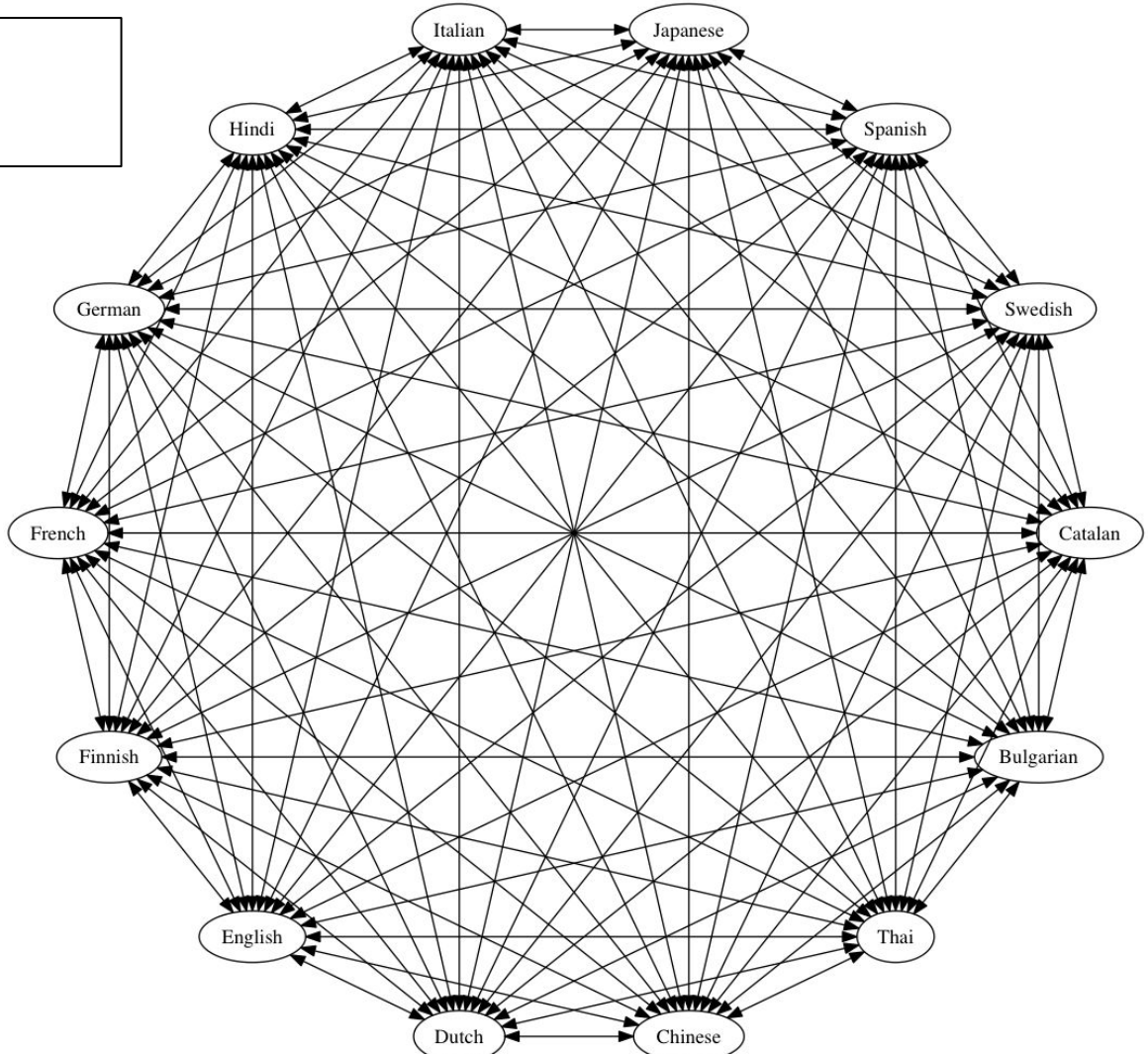
```
lin
  UttS s = s ;
  UttQS s = s ;
  UttNP np = {s = np.s ! Acc} ;
  UttAdv adv = adv ;
  UttImpSg pol imp = {s = pol.s ++ imp.s ! pol.isTrue} ;

  UseCl temp pol cl =
    let clt = cl.verb ! pol.isTrue ! temp.isPres
    in {
      s = pol.s ++ temp.s ++    --- needed for parsing
          cl.subj ++               -- she
          clt.fin ++               -- does
          negation pol.isTrue ++   -- not
          clt.inf ++               -- drink
          cl.compl                 -- beer
    } ;
```

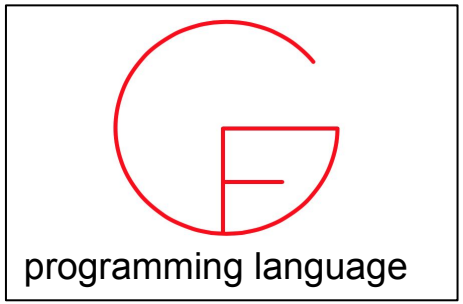
```
35    fun
36  -- Phrase
37      UttS      : S  -> Utt ;
38      UttQS     : QS -> Utt ;
39      UttNP     : NP -> Utt ;
40      UttAdv    : Adv -> Utt ;
41      UttImpSg  : Pol -> Imp -> Utt ;
42
43  -- Sentence
44      UseCl     : Temp -> Pol -> Cl
45      UseQCl    : Temp -> Pol -> QCl
46      QuestCl   : Cl -> QCl ;
47      PredVP    : NP -> VP -> Cl ;
48      ImpVP     : VP -> Imp ;
49
50  -- Verb
51      UseV      : V   -> VP ;
52      ComplV2   : V2  -> NP -> VP ;
```

Interlingual translation



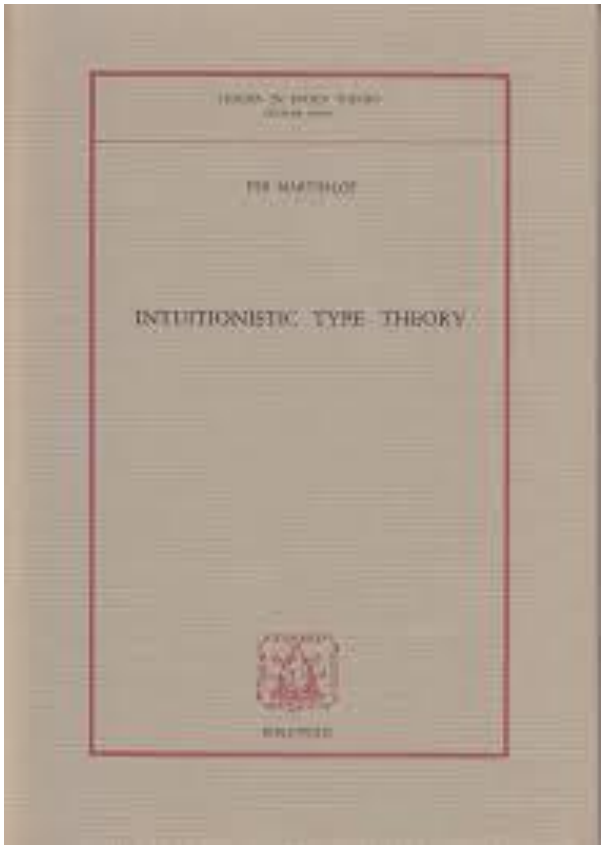
Linear scale-up

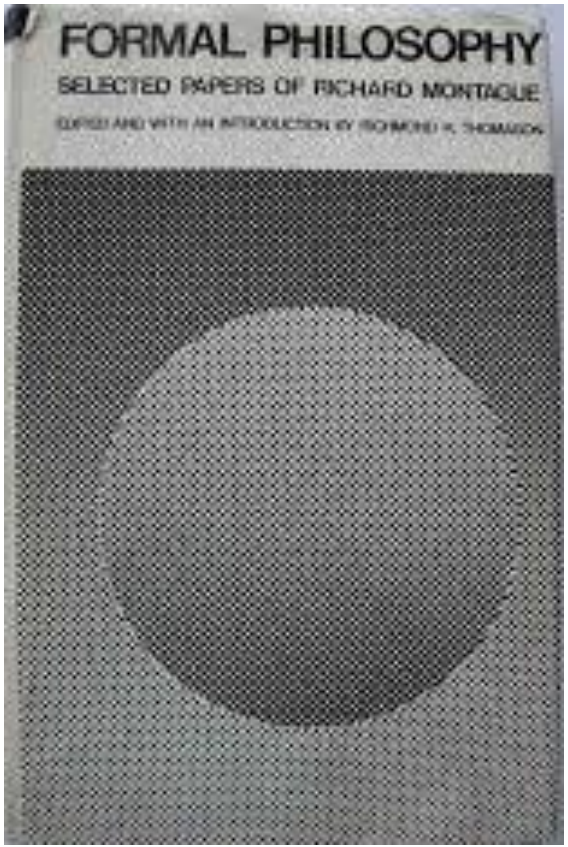


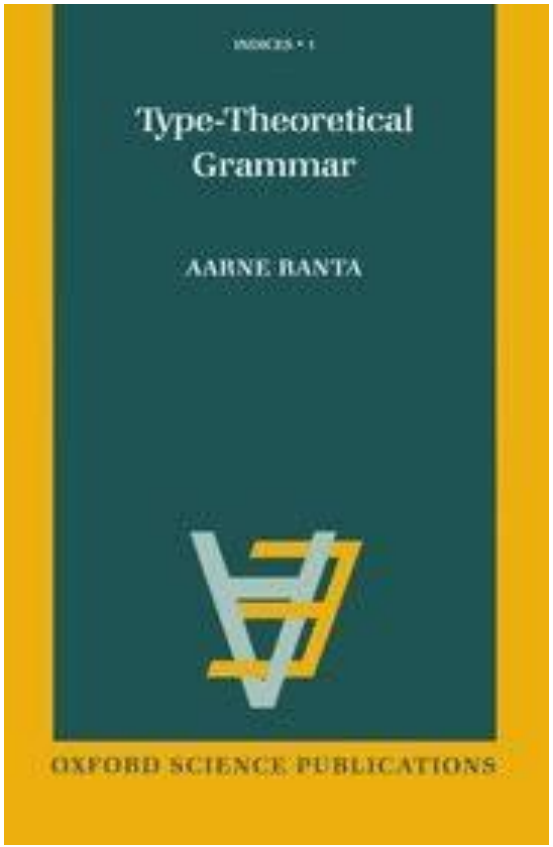


theory

Abstract syntax = Logical Framework = Type Theory







Concrete syntax = PMCFG

Thesis for the Degree of Doctor of Philosophy

Expressivity and Complexity of the Grammatical Framework

Peter Ljunglöf

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computing Science
Chalmers University of Technology and Göteborg University
SE–412 96 Göteborg, Sweden

Göteborg, November 2004

THESIS FOR THE DEGREE OF DOCTOR OF ENGINEERING

The Mechanics of the Grammatical Framework

Krasimir Angelov

CHALMERS | GÖTEBORG U

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden

Göteborg, 2011

PARALLEL MULTIPLE CONTEXT-FREE GRAMMARS, FINITE-STATE TRANSLATION SYSTEMS, AND POLYNOMIAL-TIME RECOGNIZABLE SUBCLASSES OF LEXICAL-FUNCTIONAL GRAMMARS

Hiroyuki Seki [††] Ryuichi Nakanishi [†] Yuichi Kaji [†]
Sachiko Ando [†] Tadao Kasami [††]

puter Sciences, Faculty of Engineering Science, Osaka University
neyama, Toyonaka, Osaka 560, Japan
Science, Advanced Institute of Science and Technology, Nara
kayama, Ikoma, Nara 630-01, Japan
rnet: seki@ics.es.osaka-u.ac.jp

Parallel multiple context-free grammars (pmcfg's) and *multiple context-free grammars (mcfg's)* were introduced in (Kasami 1988a)(Seki 1991) as natural extensions of cfg's. The subsystem of *lin-*

ere intro-
anguages.
ntext-free

Fast Statistical Parsing with Parallel Multiple Context-Free Grammars

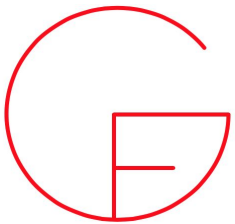
Krasimir Angelov and **Peter Ljunglöf**
University of Gothenburg and Chalmers University of Technology
Göteborg, Sweden
krasimir@chalmers.se
peter.ljunglof@cse.gu.se

Abstract

We present an algorithm for incremental statistical parsing with Parallel Multiple

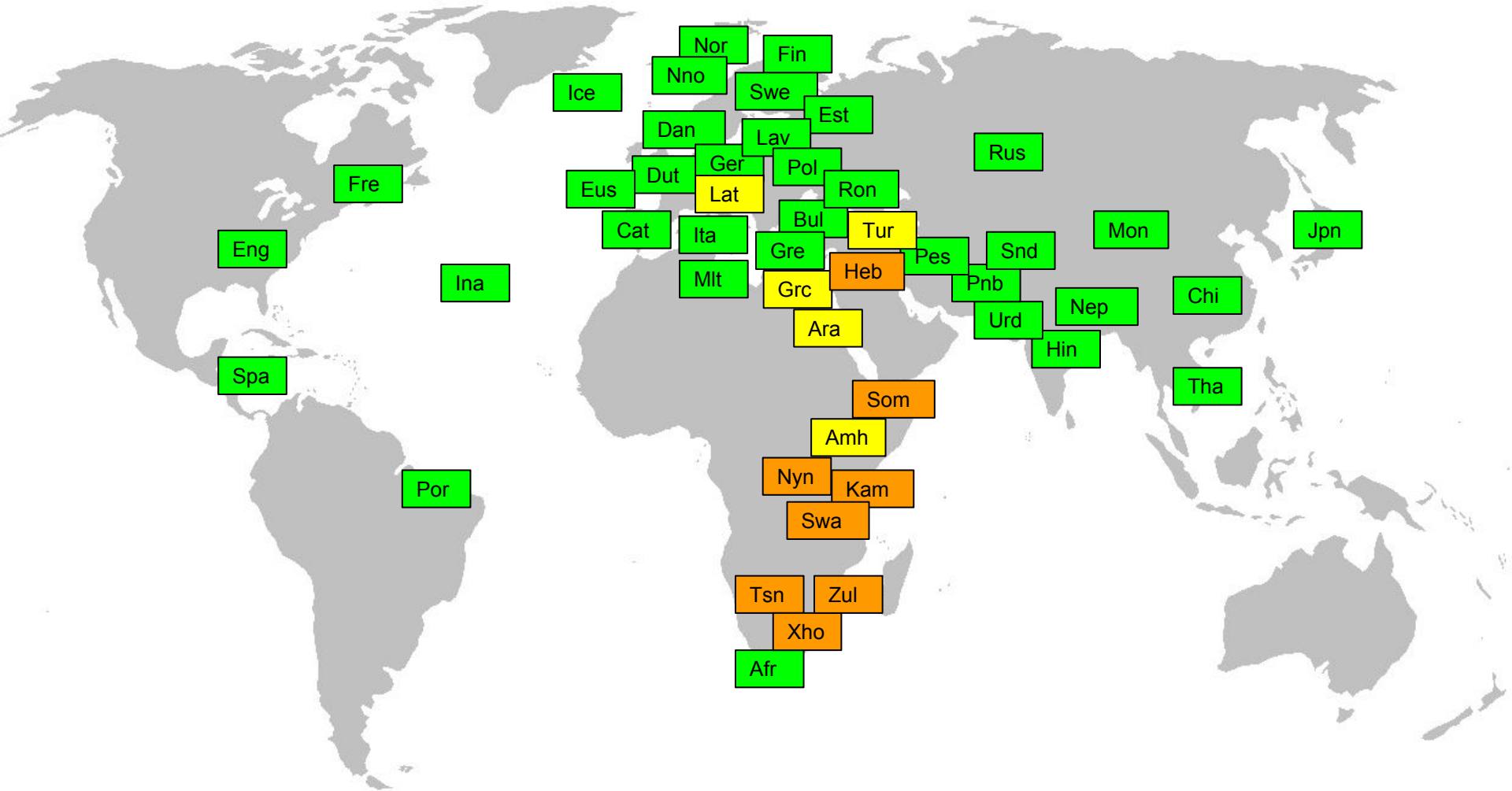
periment with novel grammar induction methods (Maier, 2013) and to use statistical disambiguation for hand-crafted grammars (Angelov, 2011).

languages



programming language

theory



https://commons.wikimedia.org/wiki/Maps_of_the_world#/media/File:BlankMap-World-noborders.png

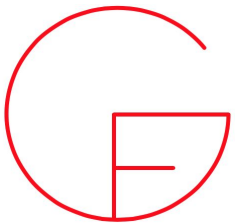
RGL = Resource Grammar Library

http://www.grammaticalframework.org/lib/doc/synopsis.html

mkCl	NP -> V2 -> NP -> Cl	*she loves him*
mkCl	NP -> V3 -> NP -> NP -> Cl	*she sen*
mkCl	NP -> VV -> VP -> Cl	*she wa*
mkCl	NP -> VS -> S -> Cl	*she say*
mkCl	NP -> VQ -> QS -> Cl	*she wo*
mkCl	NP -> VA -> A -> Cl	*she bec*
mkCl	NP -> VA -> AP -> Cl	*she bec*
mkCl	NP -> V2A -> NP -> A -> Cl	*she pai*
mkCl	NP -> V2A -> NP -> AP -> Cl	*she pai*
mkCl	NP -> V2S -> NP -> S -> Cl	*she ans*
mkCl	NP -> V2Q -> NP -> QS -> Cl	*she ask*
mkCl	NP -> V2V -> NP -> VP -> Cl	*she beg*
mkCl	NP -> VPSlash -> NP -> Cl	*she beg*
mkCl	NP -> A -> Cl	*she is o*
mkCl	NP -> A -> NP -> Cl	*she is o*
mkCl	NP -> A2 -> NP -> Cl	*she is n*
mkCl	NP -> AP -> Cl	*she is v*
mkCl	NP -> NP -> Cl	*she is t*
mkCl	NP -> N -> Cl	*she is a*
mkCl	NP -> CN -> Cl	*she is a*
mkCl	NP -> Adv -> Cl	*she is h*
mkCl	NP -> VP -> Cl	*she alw*
mkCl	N -> Cl	*there is*
mkCl	CN -> Cl	*there is*
mkCl	NP -> Cl	*there ar*
mkCl	NP -> RS -> Cl	*it is she*

- API: `mkUtt (mkCl she_NP love_V2 he_NP)`
- Afr: *sy het hom lief*
- Ara: *تُحِبُّهُ هِيَ*
- Bul: *тя го обича*
- Cat: *ella el estima*
- Chi: *她爱他*
- Dan: *hun elsker ham*
- Dut: *zij houdt van hem*
- Eng: *she loves him*
- Est: *tema armastab teda*
- Eus: *hark hura maite du*
- Fin: *hän rakastaa häntä*
- Fre: *elle l'aime*
- Ger: *sie liebt ihn*
- Gre: *αυτή τον αγαπά*
- Hin: *वह उस को प्यार करती है*
- Ice: *hún elskar hann*
- Ita: *lei lo ama*
- Jpn: *彼女は彼を愛する*
- Lav: *viņa viņu mīl*
- Mlt: *hi tħobbu*
- Mon: *түүний түүнийг хайрладаг нь*
- Nep: *उनी उ लाई माया गर्छिन्*
- Nno: *ho elskar han*
- Nor: *hun elsker ham*
- Pes: *او او را دوست دارد*
- Pnb: *او اونوں پیار کردی اے*
- Pol: *ona kocha jego*
- Por: *ela o ama*
- Ron: *ea îl iubeşte*
- Rus: *она любит его*
- Snd: *هوءِ هو سان عشق ڪري ٿي*
- Spa: *ella lo ama*
- Swe: *hon älskar honom*
- Tha: *หล่อนรักเขา*
- Urd: *وہ اس کو پیار کرتی ہے*

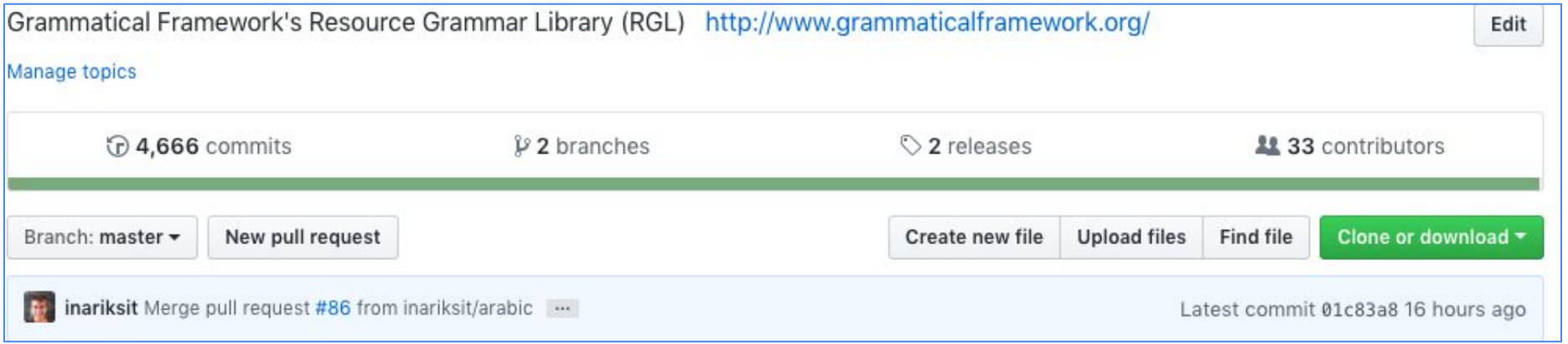
languages



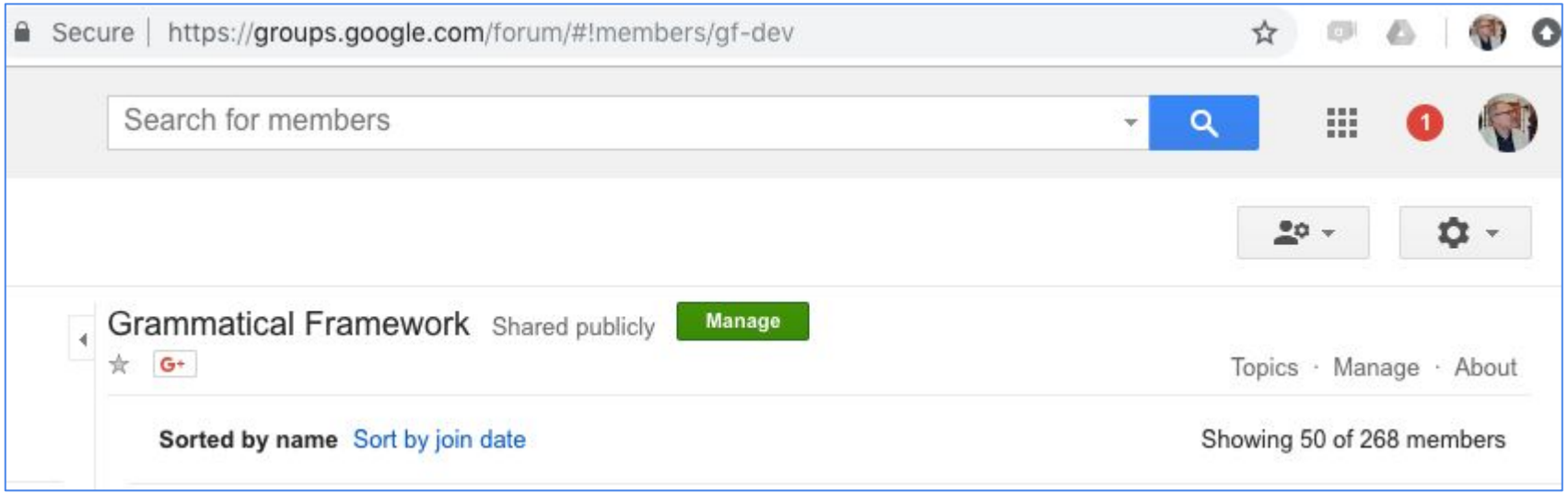
programming language

community

theory

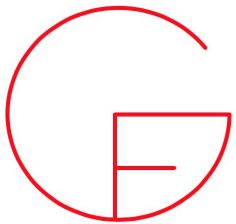


https://github.com/GrammaticalFramework



https://groups.google.com/forum/#!forum/gf-dev

languages



programming language

tools

community

theory

The GF Shell

```
Aarnes-MacBook-Pro:resource aarne$ gf

        *  *  *
      *           *
    *               *
   *
   *
   *        * * * * * *
   *        *         *
    *       * * * *  *
      *     *      *
         *  *  *

This is GF version 3.9.
Built on darwin/x86_64 with ghc-8.0, flags: interrupt server
License: see help -license.


Languages:
> i english/MiniLangEng.gf somali/MiniLangSom.gf portuguese/MiniLangPor.gf
- compiling abstract/MiniLang.gf...   write file abstract/MiniLang.gfo
- compiling english/MiniLangEng.gf...   write file english/MiniLangEng.gfo
- compiling somali/MiniLangSom.gf...   write file somali/MiniLangSom.gfo
- compiling portuguese/MiniLangPor.gf...   write file portuguese/MiniLangPor.gfo
linking ... OK

Languages: MiniLangEng MiniLangPor MiniLangSom
83 msec
MiniLang> gr -cat=S -tr | l
UseCl TAnt PNeg (PredVP (DetCN aPl_Det (UseN book_N)) (ComplV2 read_V2 (UsePron youPl_Pron)))

books have not read you
uns livros não leem vocês
buugag &+ ma ay idin akhriyaan
```

The GF Cloud

GF Cloud Service

Web Applications

- Minibar (word-completing translation tool)
- Syntax Editor (for building and manipulating abstract syntax trees)
- Translation Quiz
- GF online editor for simple multilingual grammars
- Simple Translation Tool (bilingual document editor)
- Wide Coverage Translation Demo

http://cloud.grammaticalframework.org/

APIs for embedded applications

Haskell runtime

C runtime
- Java bindings
- Python bindings
- JavaScript bindings

Android

iOS

Before you use the Python binding you need to import the pgf module.

```
>>> import pgf
```

Once you have the module imported, you can use the `dir` and `help` functio available in the object:

```
>>> dir(pgf)
```

`help` is a little bit more advanced and it tries to produce more human reada

```
>>> help(pgf)
```

A grammar is loaded by calling the method readPGF:

```
>>> gr = pgf.readPGF("App12.pgf")
```

From the grammar you can query the set of available languages. It is acces `pgf.Concr` which respresents the language. For example the following will

```
>>> eng = gr.languages["AppEng"]
>>> print(eng)
<pgf.Concr object at 0x7f7dfa4471d0>
```

Parsing

All language specific services are available as methods of the class `pgf.Co`

```
>>> i = eng.parse("this is a small theatre")
```

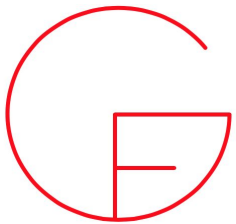
This gives you an iterator which can enumerates all possible abstract trees.

```
>>> p,e = i.next()
```

http://www.grammaticalframework.org/doc/python-api.html

applications

languages



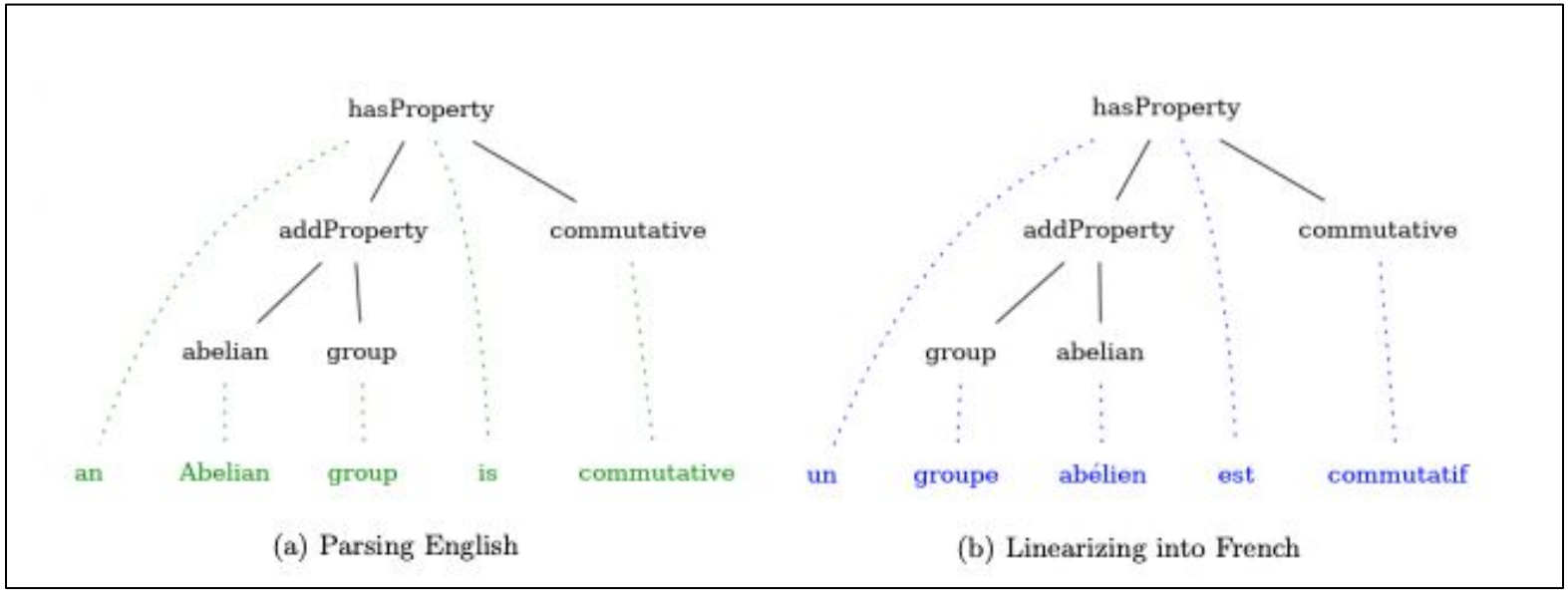
programming language

tools

community

theory

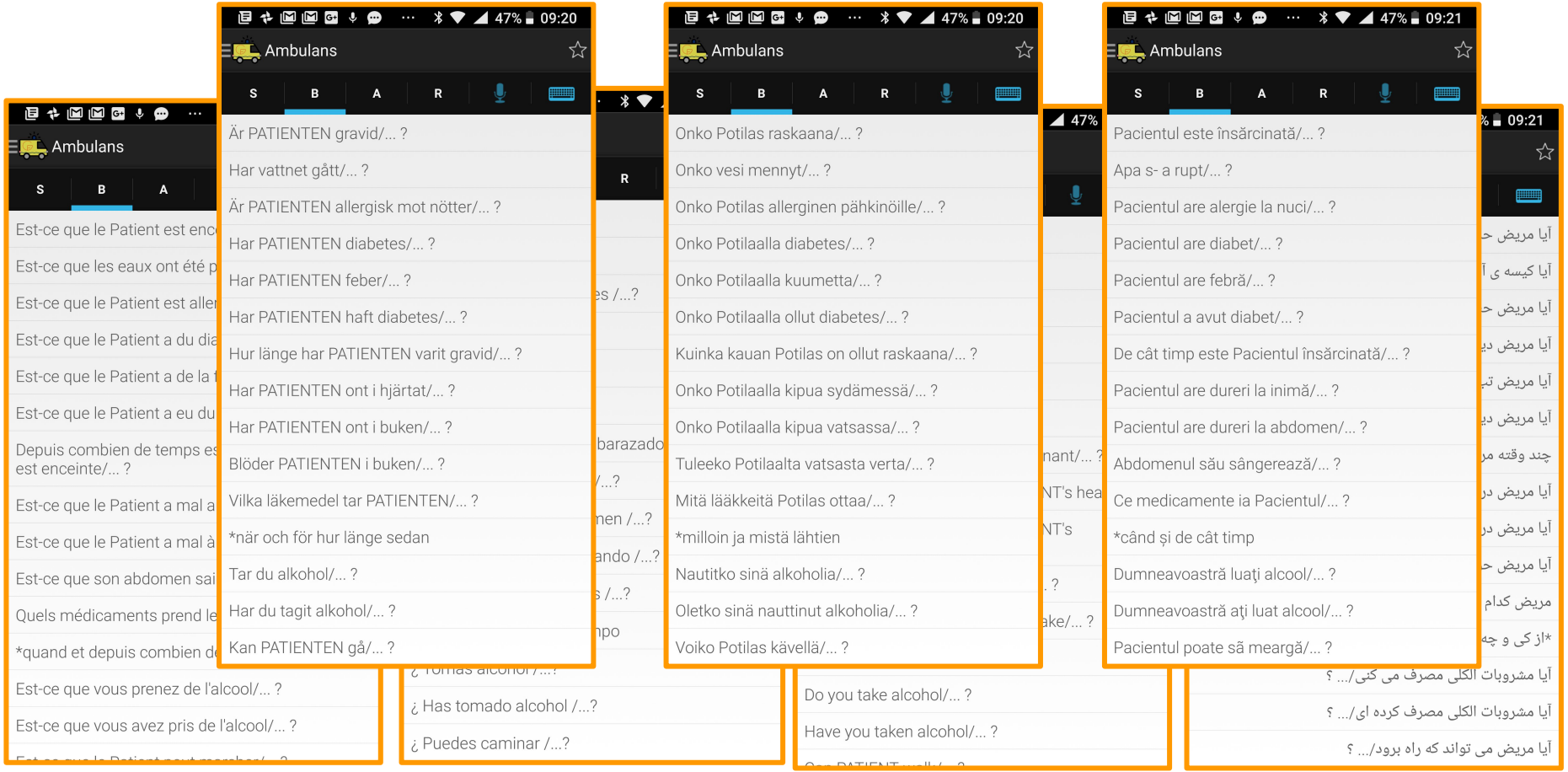
Controlled Natural Language translation



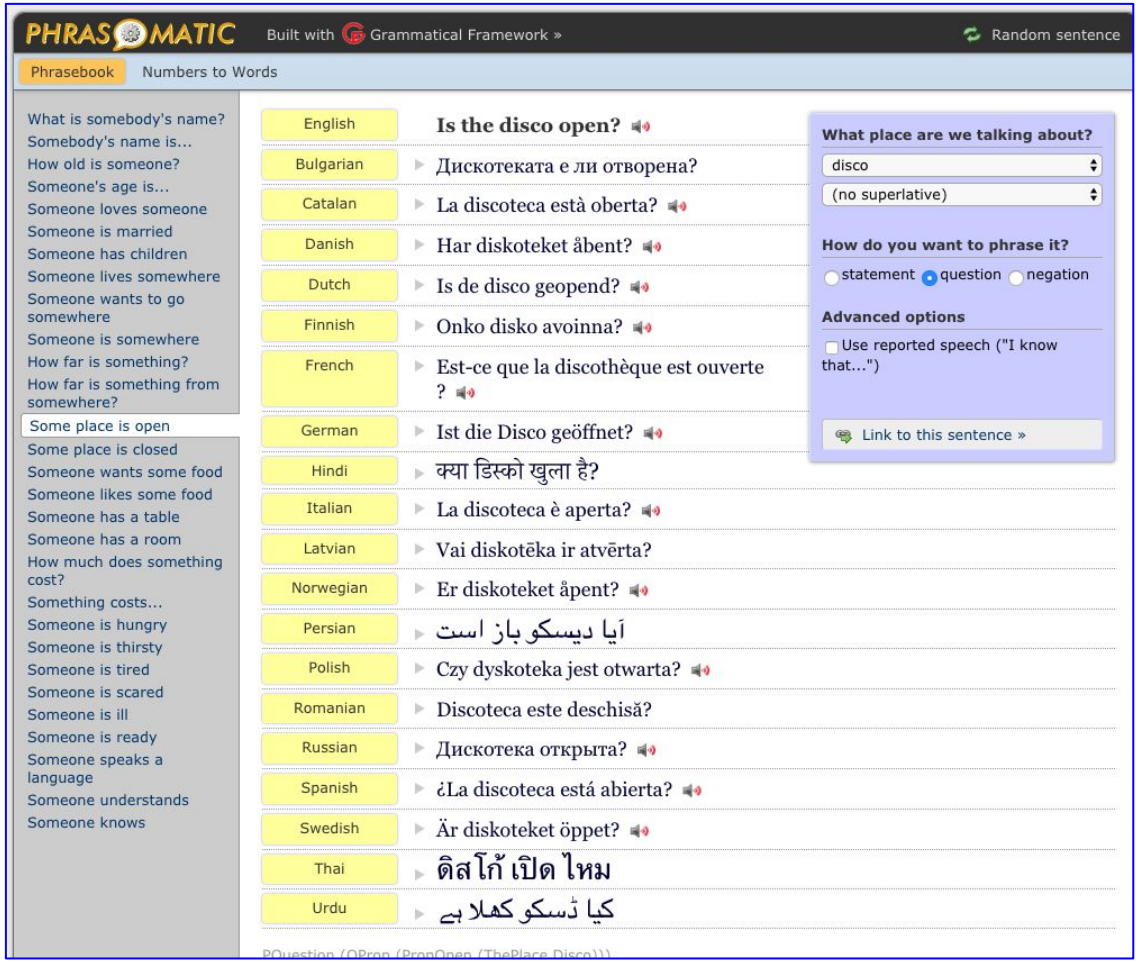
https://www.cicm-conference.org/2018/infproc/WiP1.pdf

Controlled Natural Language translation

SU Ambulans digital Grammars
Language technology to rely on.



Conceptual authoring



http://www.phrasomatic.net/

Wide-coverage translation

http://cloud.grammaticalframework.org/wc.html

Multilingual concept analysis

SIGNATU

digital Grammars
Language technology to rely on.

9 results for "subject"

English	German	French	Italian	Spanish
be subject to X	X unterliegen	être soumise à X	essere sottoposta a X	ser sujeta a X
be subject to X	X unterstehen	être placée sous les ordres de X	sottostare X	ser sujeta a X
be subject to X	X unterworfen sein	faire l' objet de X	essere soggiogata a X	ser sujeta a X
person subject to law	Rechtsunterworfene	justiciable	persona soggetta alla legge	persona sujeta al ley
subject	betroffen	concerné	soggetto	sujeto
subject	Gegenstand	sujet	soggetto	objeto
subject-matter	Gegenstand	objet	oggetto	objeto
subject to X	vorbehaltlich X	sous reserve de X	soggetto a X	siempre que se den X
subject to X	fallend unter X	soumis à X	soggetto a X	interesados a X

https://gdprlexicon.com

Natural Language Generation

UpdateOrderCommonChangeInProcessUnitTS

TSPartialOp

archTrackingData!? : optional ArchTrackingData
order!? : ORDER

---

archTrackingData!? = nil ⇒ wsInProcessUnit′ = wsInProcessUnit
archTrackingData!? ≠ nil
∧ (the archTrackingData!?).trackUnit = undefinedUnitId ⇒
wsInProcessUnit′ = {order!?} ⩤ wsInProcessUnit
archTrackingData!? ≠ nil
∧ (the archTrackingData!?).trackUnit ≠ undefinedUnitId ⇒
wsInProcessUnit′ = wsInProcessUnit ⊕
{order!? ↦ (the archTrackingData!?).trackUnit }

If the tracking data is not available, then the unit in process will not change.
If the tracking data is available, then

- if the track unit is undefined, then this order is removed from the set of orders in process.
- if the track unit is well-defined, then the order is related to the track unit in the set of orders in process.

Language learning

https://github.com/MUSTE-Project

Question answering

https://link.springer.com/chapter/10.1007%2F978-3-030-00072-1_8

3.6 Relative Clause

Below are examples of common rules building relative clauses in the GQA grammar. Words belonging to elements of relative clauses are in square brackets:

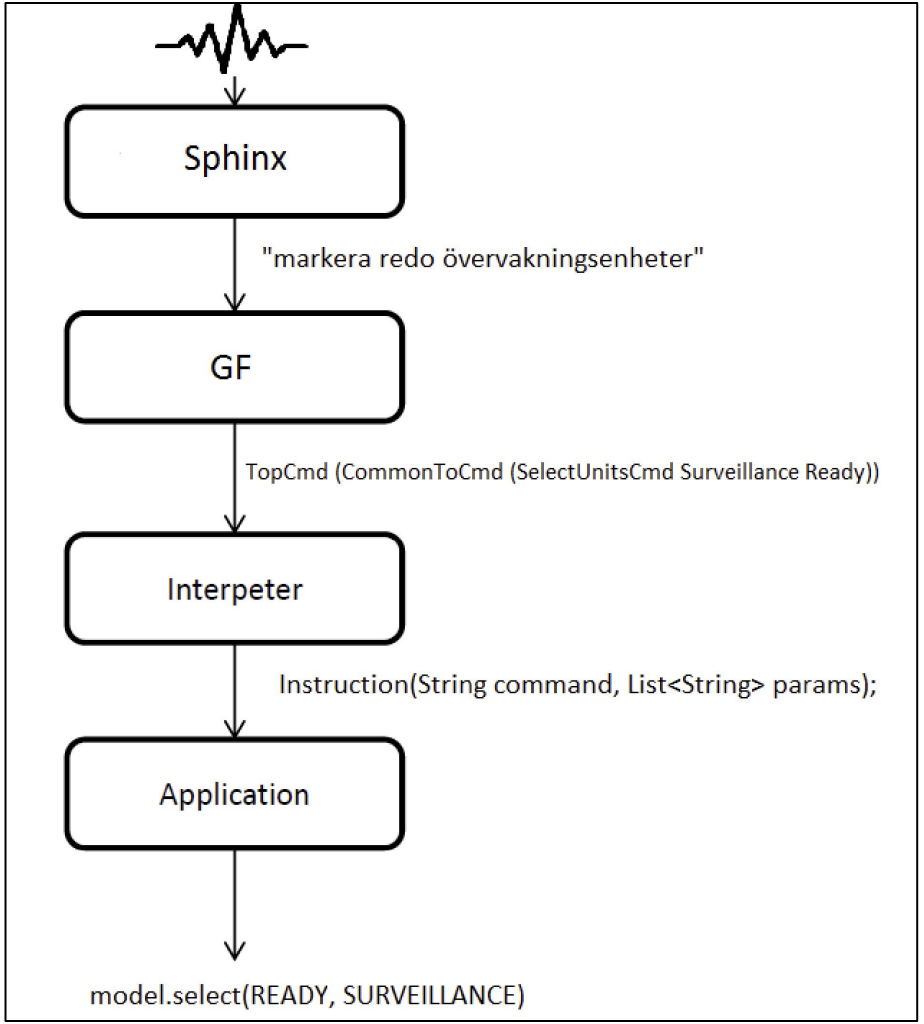
- *who/that/which + verb phrase*
  `WhoVP_relcl : VPChunk -> RelCl`
  Who are the people ***who [influenced the writers of Evenor]****?*
- *whose + property + is/are + entity*
  `WhosePropIsX_relcl : Property -> Entity -> RelCl`
  What is the city ***whose [mayor] is [Giorgos Kaminis]****?*
- *whose + property + verb phrase*
  `WhosePropVP_relcl : Property -> VPChunk -> RelCl`
  List the movies ***whose [editors] [are born in London]****.*
- *where + entity + verb phrase*
  `WhereXVP_relcl : Entity -> VPChunk -> RelCl`
  Where was the battle fought ***where [2nd Foreign Infantry Regiment] [participated]****?*

3.7 Complex Entity

Some complex constructions can perform the grammatical functions of entities. The most important of them in the GQA grammar are:

- homogeneous simple entities connected with the conjunction *and*. They are built through the rule:
  `EntityAndEntity : Entity -> Entity -> Entity ;`
  so that the corresponding branch of the tree for the question *In which team did* ***Dave Bing and Ron Reed*** *started their basketball career?* looks like:
  `EntityAndEntity (twoWordEnt "Dave" "Bing") (twoWordEnt "Ron" "Reed"),`
- a property of an entity:
  `PropOfEnt_to_Entity : Property -> Entity -> Entity`
  What is the alma mater of ***the successor of F. A. Little, Jr.****?*
  `PropOfEnt_to_Entity successor_0 (fourWordEnt "F." "A." "Little," "Jr.")`
- a simple entity followed by some property:
  `EntProp_to_Entity : Entity -> Property -> Entity`
  Name ***Ivanpah Solar power facility owner****.*
  `EntProp_to_Entity (fourWordEnt "Ivanpah" "Solar" "power" "facility") owner_0.`
- a class with a relative clause:
  `Class_to_Ent : ClassChunk -> RelCl -> Entity`

Dialogue systems

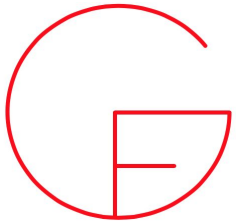


http://studentarbeten.chalmers.se/publication/253034-interpreting-voice-commands-in-radar-systems-recognizing-speech-of-multiple-languages-with-limited-r

companies

applications

languages



programming language

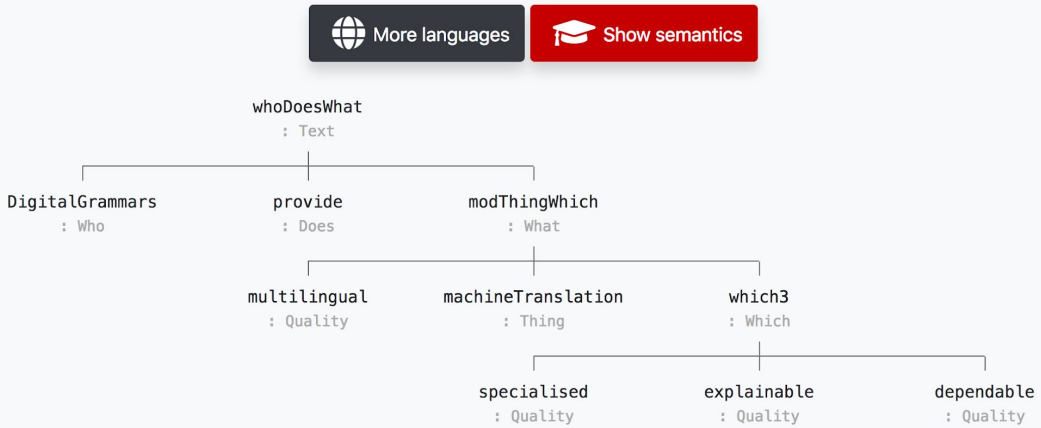
tools

community

theory

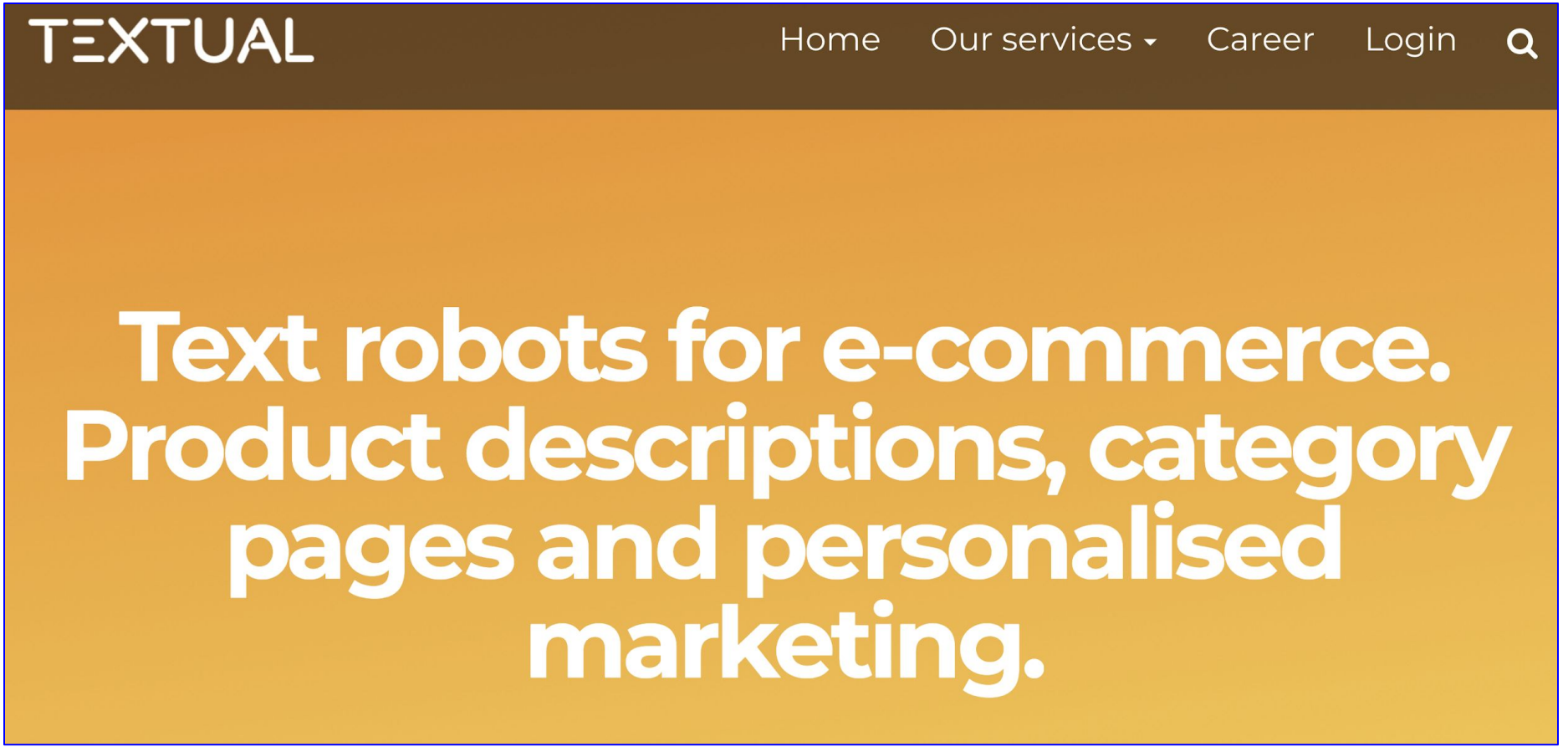
digital Grammars

Language technology to rely on.



Digital Grammars provides multilingual machine translation that is specialised, explainable and dependable.

https://www.digitalgrammars.com/



https://www.textual.se/

Talkamatic

NEXT GENERATION OF CONVERSATION

Talkamaticdevelops conversational interfaces integrating voice, chat-bots, GUI, or any other sensory system or IoT input and output.

Our TDM™system delivers advanced natural dialogue with multiple language support. We can take dialogue interfaces far beyond any other commercially available system.

TDM can be integrated into currently available systems such as Amazon Alexa, Google Assistant, Microsoft Cortana etc. to enable faster and and more cost effective development of conversational interfaces and digital assistants.

Our development is led by StaffanLarsson, Professor in Computational Linguistics at the University of Gothenburg, ensuring us to stay in the forefront of research in any area related to dialogue based human-computer interaction.

We can help you either you are completely new to the area or already have realized the limits of other solutions on the market and appreciate the challenge in creating a well functioning dialogue between humans and computers. Our consultancy services include development for any existing platform in speech or chat, as well as strategic services, research, and development.

http://www.talkamatic.se/

mercury.ai

SOLUTIONS PLATFORM COMPANY TEAM CONTACT

Solutions

Create conversational agents for dialogues that matter. Build for any use case or channel with the Mercury.ai end-to-end enterprise-grade chatbot platform. Connect to your data, your people and your third party services. In the cloud or on-premises. Measure and analyze your agents performance and your customers voice. Compliant to European standards. Build and iterate maintainable bots with our user friendly Conversation Creator. Unit test in dev - A/B test in live. Get the user right with the Mercury.ai hybrid Natural Language Understanding component. Quick and precise.

Build bots instead of Apps and utilize contextual knowledge over complete dialogues.

Use conversational AI to power your customer support

Enhance your products with conversational interfaces

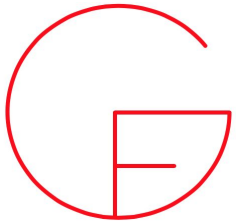
https://www.mercury.ai/#services

https://galois.com/	technology for systems where failure is unacceptable
https://www.lingsoft.fi	full-service language management company
https://www.ontotext.com/	semantic technology developer
https://www.altran.com/	world leader in engineering solutions and R&D
https://signatu.com/	support digital businesses in a legal, transparent and secure manner
http://www.lexcogitans.com/	help lawyers analyze complex legal logic

companies

applications

languages



programming language

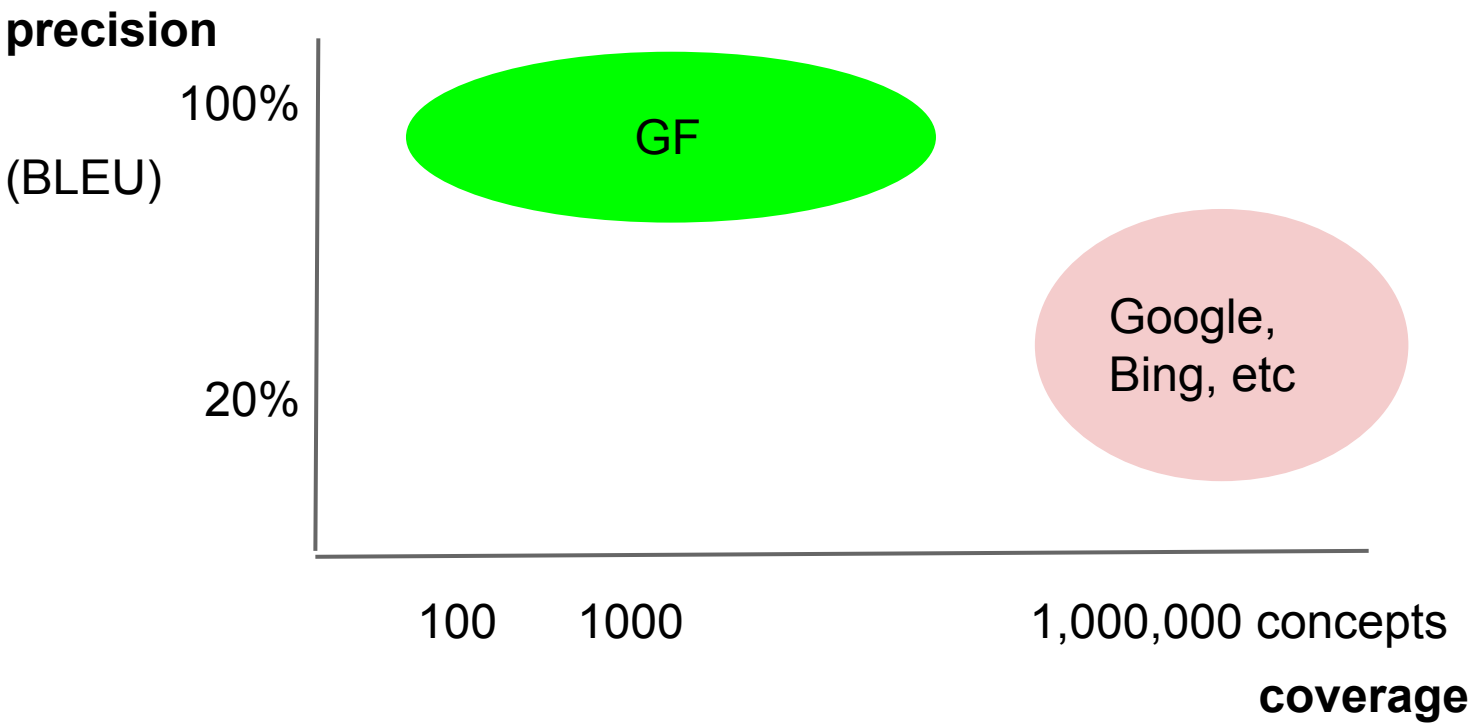
tools

community

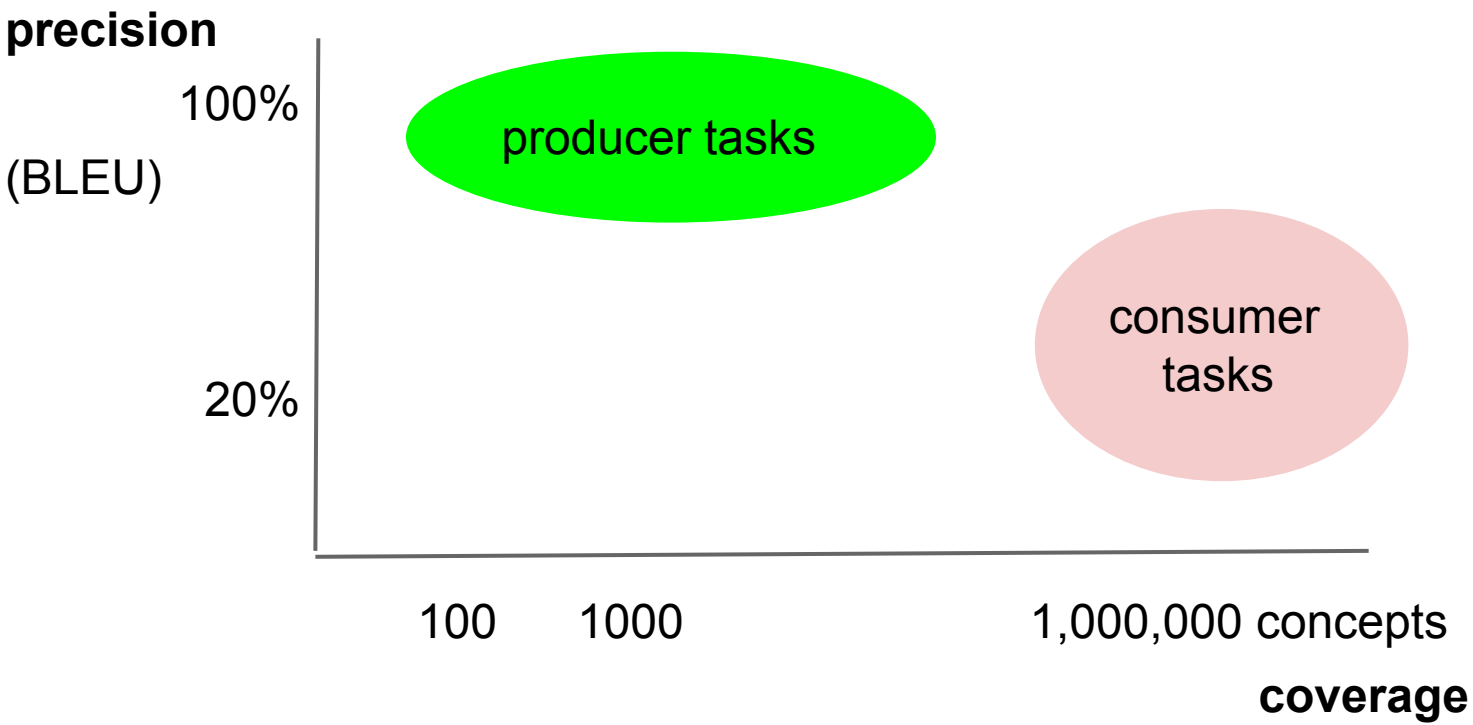
theory

research

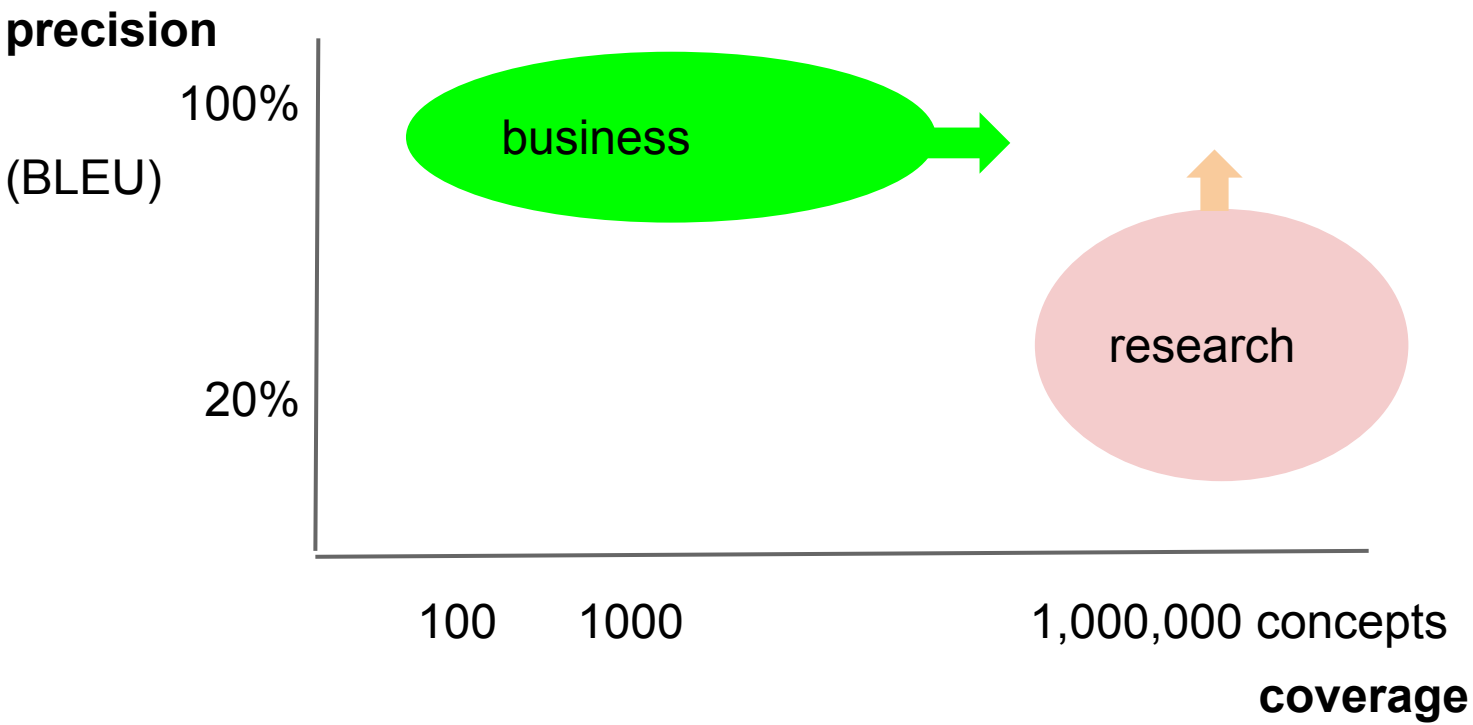
Precision vs. coverage



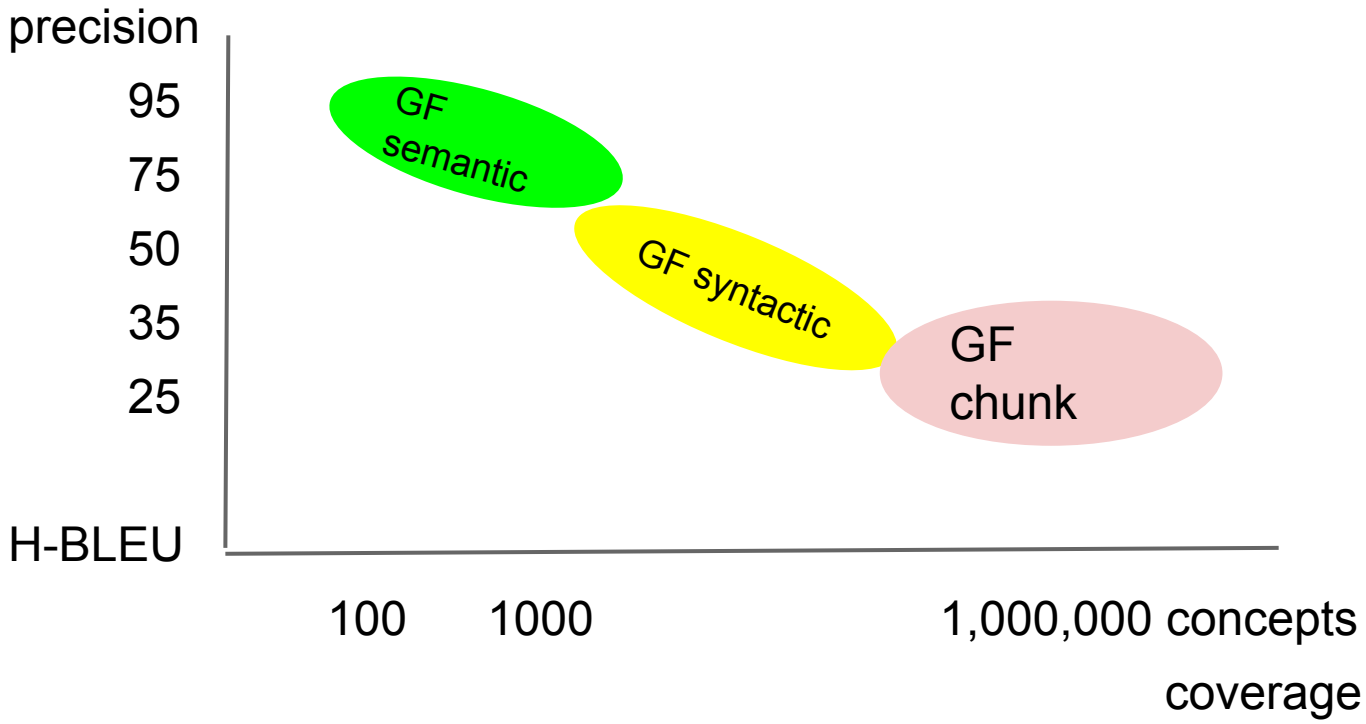
Producer vs. consumer



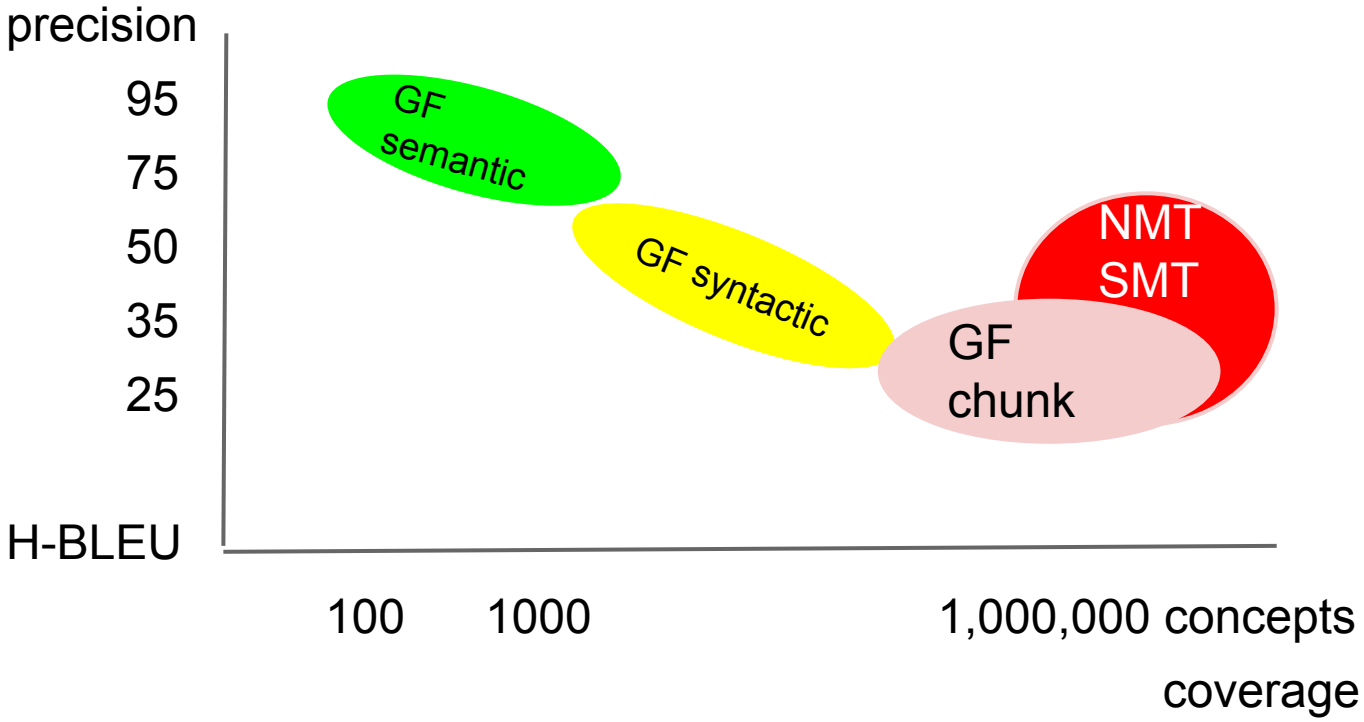
Business vs. research



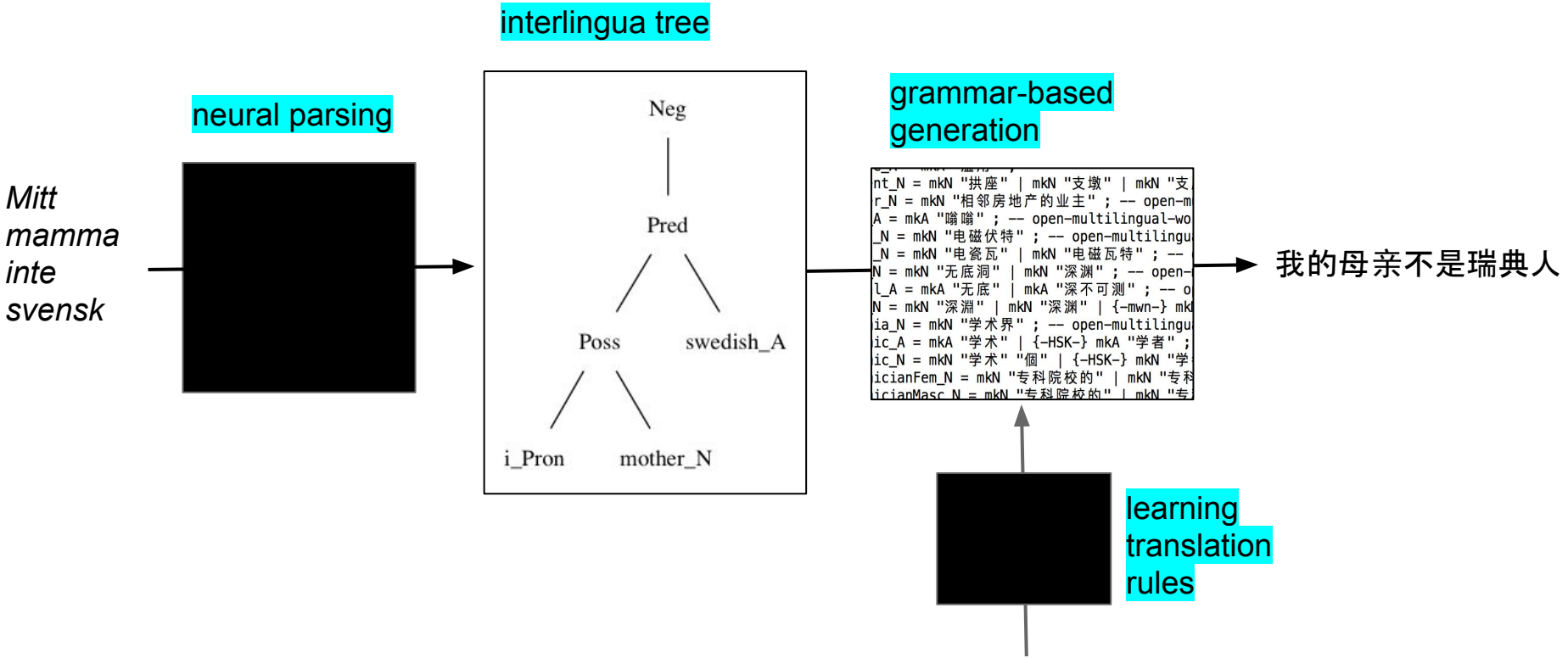
Graceful degradation



Graceful degradation ?



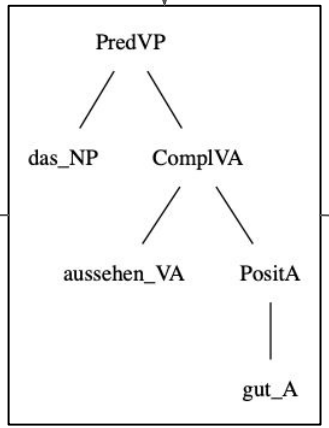
GF + black box machine learning



Data augmentation with grammar

neural translation

parsing



grammar-based generation

```
Pres Anter (CNeg False) (ODir False) : this has not looked good
Pres Anter (CNeg False) (ODir True) : this's not looked good
Pres Anter (CNeg False) OQuest : has this not looked good
Pres Anter (CNeg True) (ODir False) : this hasn't looked good
Pres Anter (CNeg True) (ODir True) : this's not looked good
Pres Anter (CNeg True) OQuest : hasn't this looked good
Past Simul CPos (ODir False) : this looked good
Past Simul CPos (ODir True) : this looked good
Past Simul CPos OQuest : did this look good
Past Simul (CNeg False) (ODir False) : this did not look good
Past Simul (CNeg False) (ODir True) : this did not look good
Past Simul (CNeg False) OQuest : did this not look good
Past Simul (CNeg True) (ODir False) : this didn't look good
Past Simul (CNeg True) (ODir True) : this didn't look good
Past Simul (CNeg True) OQuest : didn't this look good
Past Anter CPos (ODir False) : this had looked good
Past Anter CPos (ODir True) : this'd looked good
Past Anter CPos OQuest : had this looked good
Past Anter (CNeg False) (ODir False) : this had not looked good
Past Anter (CNeg False) (ODir True) : this'd not looked good
Past Anter (CNeg False) OQuest : had this not looked good
Past Anter (CNeg True) (ODir False) : this hadn't looked good
Past Anter (CNeg True) (ODir True) : this'd not looked good
Past Anter (CNeg True) OQuest : hadn't this looked good
Fut Simul CPos (ODir False) : this will look good
Fut Simul CPos (ODir True) : this'll look good
Fut Simul CPos OQuest : will this look good
Fut Simul (CNeg False) (ODir False) : this will not look good
Fut Simul (CNeg False) (ODir True) : this'll not look good
Fut Simul (CNeg False) OQuest : will this not look good
Fut Simul (CNeg True) (ODir False) : this won't look good
```

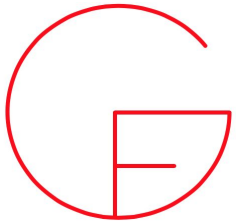
```
MIndic Cond Simul Neg Main : das würde nicht gut aussehen
MIndic Cond Simul Neg Inv : würde das nicht gut aussehen
MIndic Cond Simul Neg Sub : das nicht gut aussehen würde
MIndic Cond Anter Pos Main : das würde gut ausgesehen haben
MIndic Cond Anter Pos Inv : würde das gut ausgesehen haben
MIndic Cond Anter Pos Sub : das gut ausgesehen haben würde
MIndic Cond Anter Neg Main : das würde nicht gut ausgesehen haben
MIndic Cond Anter Neg Inv : würde das nicht gut ausgesehen haben
MIndic Cond Anter Neg Sub : das nicht gut ausgesehen haben würde
MConjunct Pres Simul Pos Main : das sehe gut aus
MConjunct Pres Simul Pos Inv : sehe das gut aus
MConjunct Pres Simul Pos Sub : das gut aussehe
MConjunct Pres Simul Neg Main : das sehe nicht gut aus
MConjunct Pres Simul Neg Inv : sehe das nicht gut aus
MConjunct Pres Simul Neg Sub : das nicht gut aussehe
MConjunct Pres Anter Pos Main : das habe gut ausgesehen
MConjunct Pres Anter Pos Inv : habe das gut ausgesehen
MConjunct Pres Anter Pos Sub : das gut ausgesehen habe
MConjunct Pres Anter Neg Main : das habe nicht gut ausgesehen
MConjunct Pres Anter Neg Inv : habe das nicht gut ausgesehen
MConjunct Pres Anter Neg Sub : das nicht gut ausgesehen habe
MConjunct Past Simul Pos Main : das sähe gut aus
MConjunct Past Simul Pos Inv : sähe das gut aus
MConjunct Past Simul Pos Sub : das gut aussähe
MConjunct Past Simul Neg Main : das sähe nicht gut aus
MConjunct Past Simul Neg Inv : sähe das nicht gut aus
MConjunct Past Simul Neg Sub : das nicht gut aussähe
MConjunct Past Anter Pos Main : das hätte gut ausgesehen
```

companies

applications

interfaces to other formats

languages



programming language

tools

community

theory

research

WordNet

```
fun deliver_2_V : V ;                -- 01440941-v        [unchecked, post] bring to a d
fun deliver_4_V : V ;                -- 02556841-v        [unchecked] free from harm or
fun deliver_8_V : V ;                -- 02557299-v        [unchecked, religion] save fro
fun deliver_1_V2 : V2 ;              -- 00991634-v        [unchecked] deliver (a speech,
fun deliver_3_V2 : V2 ;              -- 02298282-v        [unchecked] to surrender somec
fun deliver_4_V2 : V2 ;              -- 02556841-v        [unchecked] free from harm or
fun deliver_5_V2 : V2 ;              -- 02509014-v        [unchecked] hand over to the a
fun deliver_6_V2 : V2 ;              -- 01064275-v        [unchecked, law] pass down; "d
fun deliver_7_V2 : V2 ;              -- 00991901-v        [unchecked] utter (an exclamat
fun deliver_8_V2 : V2 ;              -- 02557299-v        [unchecked, religion] save fro
fun deliver_9_V2 : V2 ;              -- 02363326-v        [unchecked] carry out or perfo
fun deliver_10_V2 : V2 ;             -- 02240011-v        [unchecked] relinquish possess
fun deliver_11_V2 : V2 ;             -- 01511711-v        [unchecked] throw or hurl from
fun deliver_12_V2 : V2 ;             -- 00056644-v        [unchecked, animals, biology]
fun deliverable_A : A ;              -- 02721174-a        [unchecked] suitable for or re
fun deliverable_N : N ;              -- 03178402-n        [unchecked] something that can
and hardware"
fun deliverance_N : N ;              -- 00094303-n        [unchecked] recovery or preser
s"
fun deliverer_1_N : N ;              -- 11103646-n        [unchecked, religion] a teache
nity (circa 4 BC - AD 29)
fun deliverer_2_N : N ;              -- 10573233-n        [unchecked, person] a person w
fun deliverer_3_N : N ;              -- 10020810-n        [unchecked, commerce, person]
fun deliverer_4_N : N ;              -- 10020651-n        [unchecked, economy] a person
fun delivery_1_N : N ;               -- 00318033-n        [unchecked, commerce] the act
fun delivery_2_N : N ;               -- 07335222-n        [unchecked] the event of givin
fun delivery_3_N : N ;               -- 07085523-n        [unchecked, art, linguistics]
fun delivery_4_N : N ;               -- 01110658-n        [unchecked, economy] the volun
fun delivery_5_N : N ;               -- 00107092-n        [unchecked, baseball]  the act
fun delivery_6_N : N ;               -- 00094303-n        [unchecked] recovery or preser
fun delivery_7_N : N ;               -- 00043279-n        [unchecked, anatomy, medicine]
```

https://github.com/GrammaticalFramework/gf-wordnet

FrameNet

https://arxiv.org/pdf/1511.03924.pdf

Verb	Voice	Arguments	Freq.	Verb	Voice	Arguments	Freq.
V2	Act	NP_{dobj} NP_{nsubj}	277	V	Act	Adv Adv Adv NP_{nsubj}	2
V	Act	Adv NP_{nsubj}	155	V2	Act	Adv Adv NP_{dobj} NP_{nsubj}	2
V2	Pass	$NP_{nsubjpass}$	84	V3	Act	NP_{iobj} NP_{nsubj}	2
V2	Act	Adv NP_{dobj} NP_{nsubj}	80	VQ	Act	QS	2
V	Act	NP_{nsubj}	78	VS	Act	Adv NP_{nsubj} S	2
V2	Pass	Adv $NP_{nsubjpass}$	34	V2	Pass	Adv Adv $NP_{nsubjpass}$	2
VS	Act	NP_{nsubj} S	29	V2	Pass	Adv NP_{dobj} $NP_{nsubjpass}$	2
VV	Act	NP_{nsubj} VP	21	V2	Pass	NP_{dobj}	2
V2	Pass	NP_{dobj} $NP_{nsubjpass}$	19	V2	Act	Adv Adv NP_{dobj}	1
V2	Act	NP_{dobj}	17	V2S	Act	NP_{dobj} NP_{nsubj} S	1
V	Act	Adv Adv NP_{nsubj}	16	V2S	Act	NP_{dobj} S	1
VQ	Act	NP_{nsubj} QS	10	V2V	Act	NP_{dobj} VP	1
V2	Act	Adv NP_{dobj}	9	VS	Act	S	1
V	Act	Adv	8	VV	Act	VP	1
V2V	Act	NP_{dobj} NP_{nsubj} VP	5	V2	Pass	Adv	1
VS	Pass	S	3	VS	Pass	$NP_{nsubjpass}$ S	1

Table 8 Syntactic valence patterns underlying the shared semantico-syntactic patterns. The order of arguments (FEs) is not taken into account.

The remaining less than 7% of the shared frame functions represent the use of other verb types – VS, VQ, V2V, V3 and V2S – for which the respective RGL constructors are applied:

mkVP : VS → S → VP
$[I]_{\text{Cognizer/NP}}$ *do* $[REMEMBER]_{\text{VS}}$ $[\textit{we did a few gigs}]_{\text{Content/S}}$

mkVP : VQ → QS → VP
$[he]_{\text{Cognizer/NP}}$ $[RECOGNIZED]_{\text{VS}}$ $[\textit{where he was}]_{\text{Phenomenon/QS}}$

mkVP : V2V → NP → VP → VP
$[you]_{\text{Speaker/NP}}$ *specifically* $[REQUEST]_{\text{V2V}}$ $[me]_{\text{Addressee/NP}}$ $[\textit{to do so}]_{\text{Message/VP}}$

mkVP : V3 → NP → NP → VP
$[you]_{\text{Agent/NP}}$ $[DENIED]_{\text{V3}}$ $[her]_{\text{Protagonist/NP}}$ $[\textit{any life of her own}]_{\text{State\_of\_affairs/NP}}$

mkVP : V2S → NP → S → VP
$[he]_{\text{Speaker/NP}}$ $[PERSUADED]_{\text{V2S}}$ $[himself]_{\text{Addressee/NP}}$ $[\textit{that they helped}]_{\text{Content/S}}$

Abstract Meaning Representations (AMR)

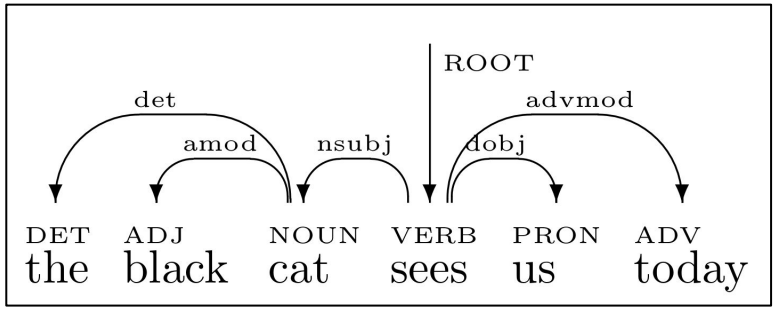
RIGOTRIO at SemEval-2017 Task 9: Combining Machine Learning and Grammar Engineering for AMR Parsing and Generation

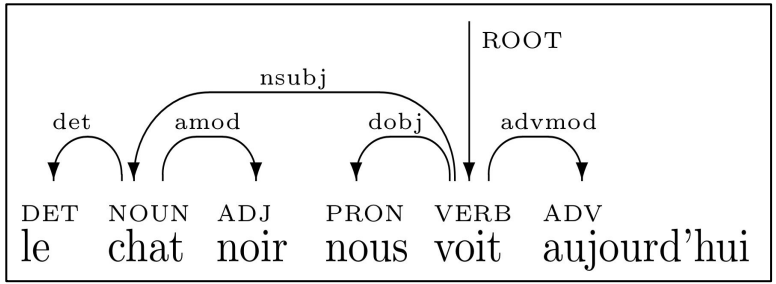
Normunds Gruzitis[1], Didzis Gosko[2] and Guntis Barzdins[1]

[1]University of Latvia, IMCS / Rainis blvd. 29, Riga, Latvia
normunds.gruzitis@lumii.lv, guntis.barzdins@lumii.lv
[2]LETA / Marija street 2, Riga, Latvia
didzis.gosko@leta.lv

http://aclweb.org/anthology/S17-2159

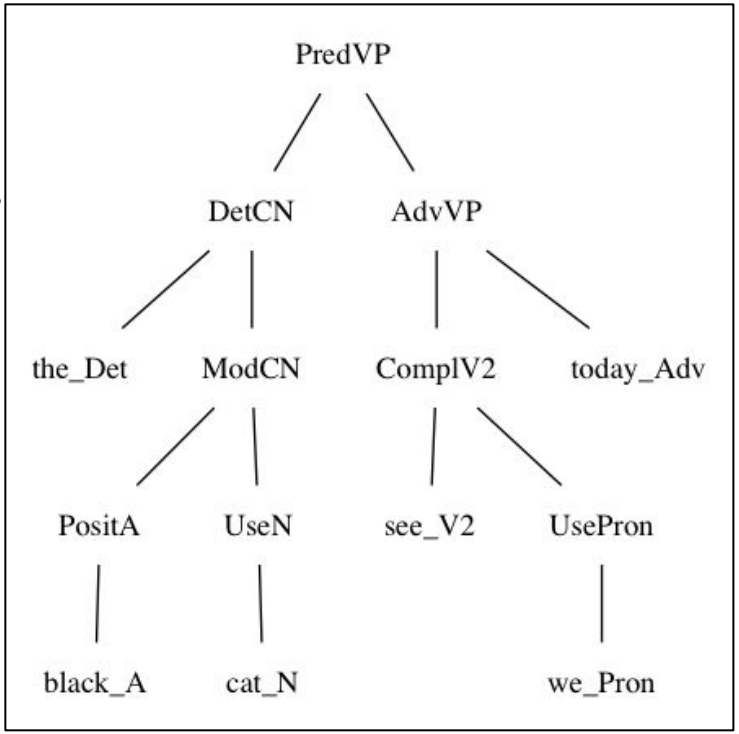
Universal Dependencies (UD)





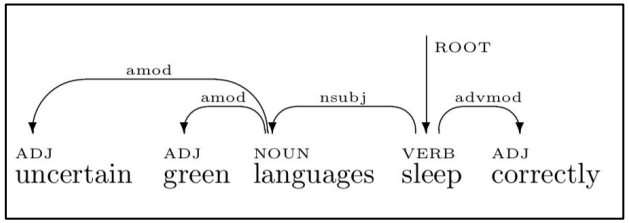
ud2gf

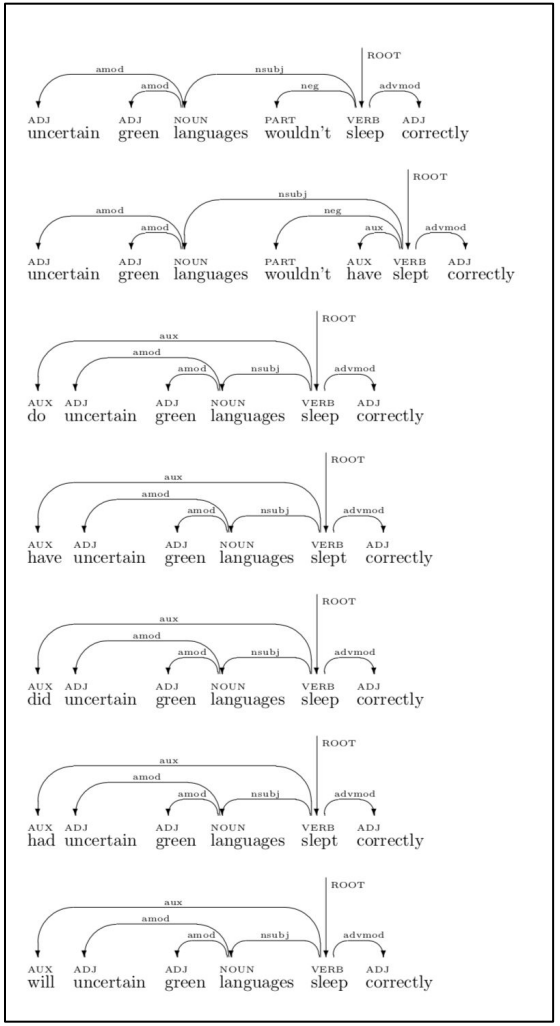
gf2ud



http://csli-lilt.stanford.edu/ojs/index.php/LiLT/article/download/71/73

UD treebank data augmentation





Xerox LEXC

```
Lexicon> pg -lexc -lang=Swa
Multichar_Symbols
+Sg +(VPast +g11_10 +AN +P3) +g11_6 +g11 +g9_10 +g7_8
VFut +Pl +Superl +AA +(AF +IN) +Compar +Posit +(VImper

LEXICON Root
enemy_N+Sg:adui # ;
swell_V+(VPast+Sg+g11_10+AN+P3):alifura # ;
swell_V+(VPast+Sg+g11_6+AN+P3):alifura # ;
swell_V+(VPast+Sg+g11+AN+P3):alifura # ;
swell_V+(VPast+Sg+g9_10+AN+P3):alifura # ;
swell_V+(VPast+Sg+g7_8+AN+P3):alifura # ;
swell_V+(VPast+Sg+g6+AN+P3):alifura # ;
swell_V+(VPast+Sg+g5a_6+AN+P3):alifura # ;
swell_V+(VPast+Sg+g5_6+AN+P3):alifura # ;
swell_V+(VPast+Sg+g3_4+AN+P3):alifura # ;
swell_V+(VPast+Sg+g1_2+IN+P3):alifura # ;
swell_V+(VPast+Sg+g1_2+IN+P2):alifura # ;
swell_V+(VPast+Sg+g1_2+IN+P1):alifura # ;
swell_V+(VPast+Sg+g1_2+AN+P3):alifura # ;
come_V+(VPast+Sg+g11_10+AN+P3):alikuja # ;
come_V+(VPast+Sg+g11_6+AN+P3):alikuja # ;
come_V+(VPast+Sg+g11+AN+P3):alikuja # ;
come_V+(VPast+Sg+g9_10+AN+P3):alikuja # ;
come_V+(VPast+Sg+g7_8+AN+P3):alikuja # ;
come_V+(VPast+Sg+g6+AN+P3):alikuja # ;
come_V+(VPast+Sg+g5a_6+AN+P3):alikuja # ;
come_V+(VPast+Sg+g5_6+AN+P3):alikuja # ;
```

Speech recognition formats

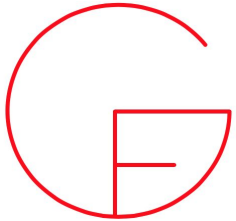
```
Doctor> pg -printer=srgs_abnf -lang=Som
#ABNF 1.0 UTF-8;
meta "description" is "Speech recognition grammar for DoctorSom";
meta "generator" is "Grammatical Framework";
root $Phrase_cat;

$Action_1 = neefi | qufaci | cabi | cuni | baarayi
            | tagi | seexn | cabi | joogi | adodan | tallaalin
            | matagi ;

$Action_10 = neeftay | qufacday | cabtay | cuntay
             | baaraysay | tagtay | seexday | cabtay | joogtay
             | adodatay | tallaalisey | matagtay ;

$Action_11 = neefnay | qufacnay | cabnay | cunnay
             | baaraynay | tagnay | seexnay | cabnay | joognay
             | adodannay | tallaalinney | matagnay ;

$Action_12 = neefteen | qufacdeen | cabteen
             | cunteen | baarayseen | tagteen | seexdeen
             | cabteen | joogteen | adodateen | tallaaliseen
             | matagteen ;

$Action_13 = neefeen | qufaceen | cabbeen | cuneen
             | baarayeen | tageen | seexeen | cabbeen | joogeen
             | adoteen | tallaaliyeen | matageen ;

$Action_14 = neefin | qufacin | cabin | cunin
             | baarayin | tagin | seexin | cabin | joogin
             | adodaan | tallaaliin | matagin ;

$Action_15 = neefto | qufacdo | cabto | cunto
             | baarayso | tagto | seexdo | cabto | joogto
             | adodato | tallaaliso | matagto ;

$Action_16 = neef | qufac | cab | cun | baaray
```

companies

applications

interfaces to other formats

languages



programming language

tools

community

theory

research

publications

Grammatical Framework

About 458,000 results (**0.08** sec)

Grammatical framework
A Ranta - Journal of Functional Programming, 2004 - cambridge.org
Grammatical Framework (GF) is a special-purpose functional language for defining grammars. It uses a Logical Framework (LF) for a description of abstract syntax, and adds to this a notation for defining concrete syntax. GF grammars themselves are purely declarative …
☆ Cited by 351 Related articles All 9 versions

[BOOK] **Grammatical framework**: Programming with multilingual grammars
A Ranta - 2011 - grammaticalframework.org
Page 1. **Grammatical Framework**: Programming with Multilingual Grammars Aarne Ranta Slides for the GF book (CSLI, Stanford, 2011) Page 2. Page 3. Preamble Page 4. GF, **Grammatical Framework** A special-purpose programming language for writing grammars. Supports the …
☆ Cited by 194 Related articles All 6 versions

Eco-grammar systems: A **grammatical framework** for studying lifelike interactions
E Csuhaj-Varjú, J Kelemen, A Kelemenová, G Păun - Artificial Life, 1997 - MIT Press
A formal **framework** for studying systems made up of a community of agents and their environment is proposed. The suggested model, technically based on the theory of formal grammars and called an eco-grammar system, captures some common features of …
☆ Cited by 127 Related articles All 14 versions

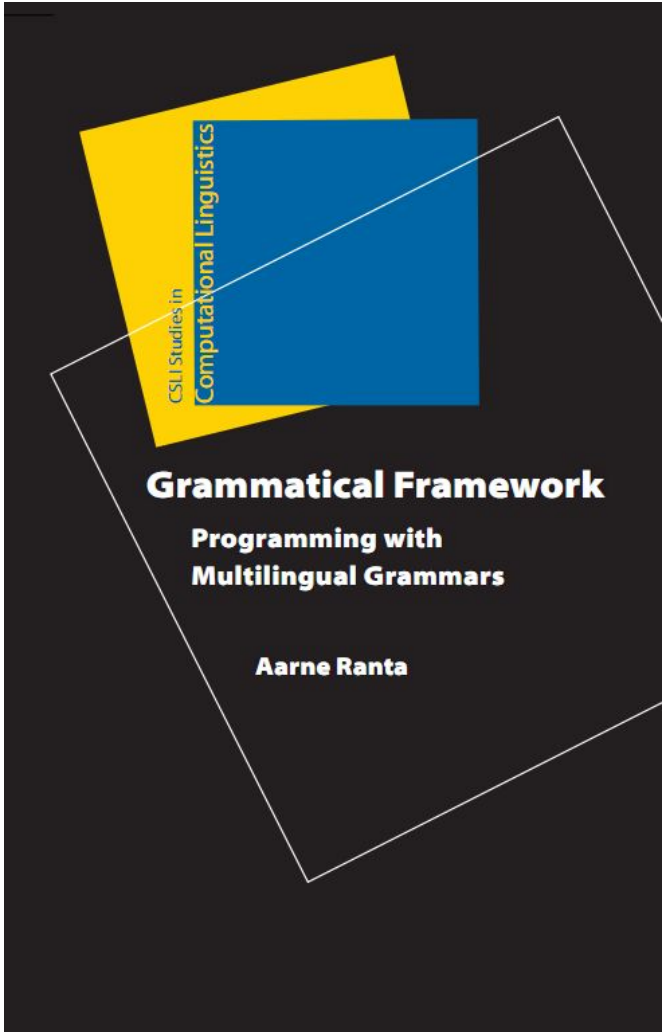
[PDF] Expressivity and complexity of the **Grammatical Framework**
P Ljunglöf - 2004 - gupea.ub.gu.se
This thesis investigates the expressive power and parsing complexity of the **grammatical framework** (gf), a formalism originally designed for displaying formal propositions and proofs in natural language. This is done by relating gf with two more well-known grammar …
☆ Cited by 54 Related articles All 5 versions

Related searches

grammatical framework **grammar**
grammatical framework **ontology**
grammatical framework **natural language processing**
ranta grammatical framework
grammatical framework **programming language**
grammatical framework **parsing**
grammatical framework **syntax**
grammatical framework **further conclusion**

A multilingual semantic wiki based on Attempto Controlled English and **Grammatical Framework**
K Kaljurand, T Kuhn - Extended Semantic Web Conference, 2013 - Springer
We describe a semantic wiki system with an underlying controlled natural language grammar implemented in **Grammatical Framework** (GF). The grammar restricts the wiki content to a well-defined subset of Attempto Controlled English (ACE), and facilitates a …
☆ Cited by 31 Related articles All 17 versions

[BOOK] The mechanics of the **grammatical framework**
K Angelov - 2011 - pdfs.semanticscholar.org
Grammatical Framework (GF) is a well known theoretical **framework** and a mature programming language for the description of natural languages. The GF community is growing rapidly and the range of applications is expanding. Within the **framework**, there are …
☆ Cited by 37 Related articles All 4 versions



http://www.grammaticalframework.org/gf-book/

Punjabi

- **Source**: http://www.grammaticalframework.org/lib/src/punjabi (Shafqat Virk, Muhammad Humayoun)
- **Publications**

  S. Virk, M. Humayoun, and A. Ranta. An Open-Source Punjabi Resource Grammar. Proceedings of RANLP-2011, Recent Advances in Natural Language Processing, Hissar, Bulg
  http://lml.bas.bg/~iva/ranlp2011/RANLR2011_Proceedings.PDF

  M. Humayoun and A. Ranta. Developing Punjabi Morphology, Corpus and Lexicon. *The 24th Pacific Asia conference on Language, Information and Computation (PACLIC24)*, 2(

Romanian

- **Source**: http://www.grammaticalframework.org/lib/src/romanian (Ramona Enache)
- **Publications**

  R. Enache, A. Ranta, and K. Angelov. An Open-Source Computational Grammar of Romanian. A. Gelbukh (ed.), *CiCLING-2010*, LNCS 6008, 2010.

Russian

- **Source**: http://www.grammaticalframework.org/lib/src/russian (Janna Khegai, Nikita Frolov)
- **Publications**

  J. Khegai. GF parallel resource grammars and Russian. In proceedings of ACL2006 (The joint conference of the International Committee on Computational Linguistics and the As
  (pp. 475-482), Sydney, Australia, July 2006.

  J. Khegai and A. Ranta. Building and Using a Russian Resource Grammar in GF. In A. Gelbukh (ed), *Intelligent Text Processing and Computational Linguistics (CICLing-2004)*, S
  LNCS 945, pp. 38-41, 2004.

Sindhi

- **Source**: http://www.grammaticalframework.org/lib/src/sindhi (Jherna Devi Oad)
- **Publications**

  Jherna Devi Oad. Implementing GF Resource Grammar for Sindhi language, MSc Thesis, Chalmers University of Technology, 2012. http://publications.lib.chalmers.se/records/ful

http://www.grammaticalframework.org/lib/doc/rgl-publications.html

Expressivity and Complexity of the Grammatical Framework

Peter Ljunglöf

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computing Science
Chalmers University of Technology and Göteborg University
SE–412 96 Göteborg, Sweden

Formal and Informal Software Specifications

Kristofer Johannisson

CHALMERS | GÖTEBORGS UNIVERSITET

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, 2005

Thesis for the Degree of Doctor of Philosophy

Language Engineering in Grammatical Framework (GF)

Janna Khegai

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computing Science
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, September 2006

Thesis for the Degree of Doctor of Engineering

Three Tools for Language Processing: BNF Converter, Functional Morphology, and Extract

Markus Forsberg

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, Sweden 2007

Programming Language Techniques for Natural Language Applications

Björn Bringert

CHALMERS | UNIVERSITY OF GOTHENBURG

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg
Sweden

Gothenburg, October 2008

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR

de l'Université de Grenoble

Spécialité: Mathématiques et Informatiques

Présentée et soutenue publiquement par

MUHAMMAD HUMAYOUN

Le 18 janvier 2012

TITRE:

Developing the System MathNat for Automatic Formalization of Mathematical texts

Thèse dirigée par **Christophe Raffalli** et **Aarne Ranta**

Préparée au sein du **LAMA, Université de Savoie** et de l'**École Doctorale MSTII**

JURY

Civilité/Nom/Prénom	Fonction et lieu de la fonction	Rôle
M Loïc POTTIER	HDR, Chargé de recherche, INRIA Sophia Antipolis	Rapporteur
M Zhaohui LUO	Professeur, University of London	Rapporteur
M Laurent VUILLON	Professeur, LAMA, Université de Savoie	Examinateur
Mme Laurence DANLOS	Professeur, Université de Paris 7	Présidente du jury
M Christophe RAFFALLI	Maître de Conférences, LAMA, Université de Savoie	Directeur de thèse
M Aarne RANTA	Professeur, University of Gothenburg	Directeur de thèse

---

Thesis for the degree of Doctor of Engineering

The Mechanics of the Grammatical Framework

Krasimir Angelov

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden

Göteborg, 2011

Multilingual text generation from structured formal representations

Dana Dannélls

Data linguistica

Thesis for the Degree of Doctor of Philosophy

Computational Linguistics Resources for Indo-Iranian Languages

Shafqat Mumtaz Virk

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science and Engineering
Chalmers University of Technology &
University of Gothenburg
Gothenburg, Sweden 2013

Frontiers of Multilingual Grammar Development

Ramona Enache

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden

Göteborg, 2013

UNIVERSITY OF LATVIA
Institute of Mathematics and Computer Science

NORMUNDS GRŪZĪTIS

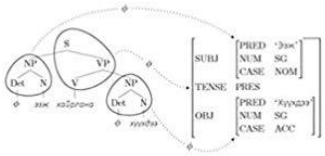
FORMAL GRAMMAR AND SEMANTICS OF CONTROLLED LATVIAN LANGUAGE

Summary of Doctoral Thesis
in Computer Science

Riga 2010

Implementierung der Grammatik des modernen Mongolischen in der funktionalen Programmiersprache „Grammatical Framework"

Nyamsuren Erdenebadrakh



sierke

Methods and tools for automating language engineering

GRÉGOIRE DÉTREZ

UNIVERSITY OF GOTHENBURG

Department of Computer Science and Engineering
Chalmers University of Technology & University of Gothenburg
Göteborg, Sweden 2016

Contracts and Computation

Formal modelling and analysis for normative natural language

JOHN J. CAMILLERI

UNIVERSITY OF GOTHENBURG

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden 2017

---

universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

Dynamic Generalized Parsing and Natural Mathematical Language

verfasst von / submitted by
Mag. Kevin Kofler, Bakk.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Doktor der Naturwissenschaften (Dr.rer.nat)

Wien, 2017 / Vienna, 2017

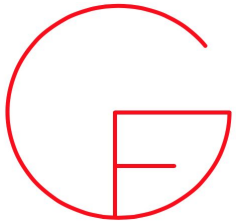
Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 091 405
Dissertationsgebiet lt. Studienblatt / field of study as it appears on the student record sheet:	Mathematik
Betreut von / Supervisor:	Univ.-Prof. Dr. Arnold Neumaier

companies

applications

interfaces to other formats

languages



programming language

tools

community

theory

Summer Schools

research

publications

GF Summer Schools

Gothenburg, Sweden, 2009

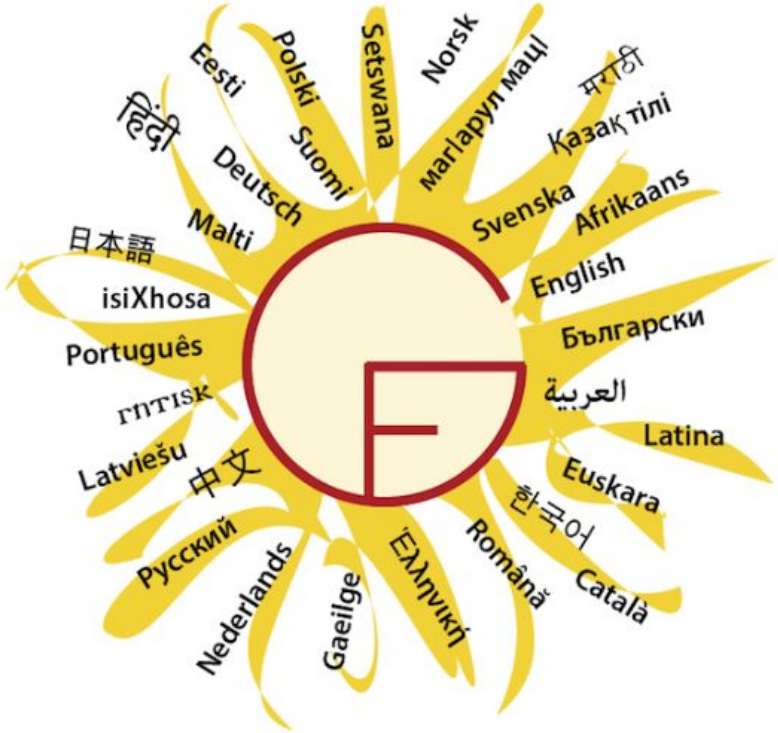
Barcelona, Catalonia, 2011

Frauenchiemsee, Bavaria, 2013

Marsalforn, Gozo, Malta, 2015

Riga, Latvia, 2017

Sixth GF Summer School 2018



Stellenbosch, South Africa

3rd–14th December 2018

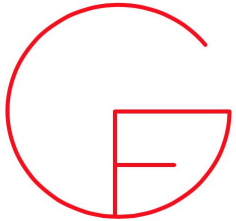
The GF Ecosystem

companies

applications

interfaces to other formats

languages



programming language

tools

community

theory

Summer Schools

research

publications