UNIVERSITY OF GOTHENBURG

Controlled Natural Language Generation from a Multilingual FrameNet-based Grammar

Dana Dannélls, Department of Swedish

Normunds Grūzītis, Department of Computer Science and Engineering

4th Workshop on Controlled Natural Language, 20–22 August 2014, Galway, Ireland

Previous and recent work

- Normunds Grūzītis, Guntis Bārzdiņš. **Polysemy in Controlled Natural Language Texts**. CNL 2009
- Dana Dannélls. **Applying semantic frame theory to automate natural language templates generation from ontology statements**. INLG 2010
- Dana Dannélls, Lars Borin. **Toward language independent methodology for generating artwork descriptions – Exploring FrameNet information**. LaTeCH 2012
- Normunds Grūzītis, Pēteris Paikens, Guntis Bārzdiņš. **FrameNet Resource Grammar Library for GF**. CNL 2012
- Normunds Grūzītis. **A frame-semantic abstraction layer to GF RGL**. GF Summer School 2013
- Dana Dannélls, Normunds Grūzītis. **Extracting a bilingual semantic grammar from FrameNet-annotated corpora**. LREC 2014

General aim

Abstract Syntax ⟷ Multilingual Concrete Syntax

NL text	Objects	FN Events	GF-EN Paraphrase	GF-LV Paraphrase
Sophie Amundsen was on her way home from school.	X1:Sophie Amundsen; X72:home; X73:school; X3:way;	E1:Self_motion(self_mover:X1; source:X73; goal:X72; path:X3)	E1:Sophie Amundsen moved from school to home.	E1:Sofija Amundsena pārvietojās no skolas uz mājām
She had walked the first part of the way with Joanna.	X4: the first part of X3; X5:Joanna;	E2: Self_motion(self_mover:X1; path:X4; co_theme:X5; time:during E1)	E2:During E1 the first part of the way Sophie Amundsen walked with Joanna.	E2: E1 laikā ceļa pirmo pusi Sofija Amundsena gāja kopā ar Jūrunu.
They had been discussing robots.	X6: robots;	E3: Discussion(interlocutors: X1,X5; topic:X6; time:during E2)	E3:During E2 Sophie Amundsen and Joanna discussed robots.	E3: E2 laikā Sofija Amundsena un Jūruna apsprieda robotus.
Joanna thought		E4:Opinion(cognizer:X5; opinion:E5; time:during E3)	E4:During E3 Joanna stated E5.	E4: E3 laikā Jūruna apgalvoja E5.
the human brain was like an advanced computer.	X7:the human brain; X8: an advanced computer;	E5: Similarity(entity1:X7; entity2:X8)	E5:The human brain is similar to an advanced computer.	E5: Cilvēka smadzenes ir līdzīgas sarežģītam datoram.

A slide from CNL 2012

Outline

- Background and the specific aim
- Extracting semantico-syntactic valence patterns from FrameNet-annotated corpora
- Generating a multilingual FrameNet-based grammar in GF
- Case studies
- Initial evaluation
- Conclusions and future work

FrameNet

- A lexico-semantic resource based on the theory of frame semantics (Fillmore et al., 2003)
  - A semantic **frame** represents a prototypical, language-independent situation characterized by **frame elements** (**FE**) – **semantic valence**
  - A frame is evoked in a sentence by a language-specific **lexical unit** (**LU**)
  - FEs are mapped based on the **syntactic valence** of the LU
    - The syntactic and semantic **valence patterns** are derived from FrameNet-annotated corpora (for an increasing number of languages)
  - FEs are divided into **core** and **non-core** ones
    - Core FEs uniquely characterize the frame and syntactically correspond to verb **arguments**
    - Non-core FEs (**adjuncts**) are not specific to the frame

BFN and SweFN

- Currently, we consider two framenets (FN): the original Berkeley FrameNet (**BFN**) and the Swedish FrameNet (**SweFN**)
  - Only frames for which there is at least one corpus example where the frame is evoked by a <u>verb</u>

- BFN 1.5 defines >1,000 frames of which **556** are evoked by ~3,200 verb LUs in >**68,500** annotated sentences
- The SweFN development version covers >900 frames of which **638** are evoked by ~2,300 verb LUs in >**3,700** sentences

- SweFN, like many other FNs, mostly reuses BFN frames, hence, BFN frames can be seen as a semantic **interlingua**

Example

Desiring	
Definition:	An Experiencer desires that an Event occur. In some cases, the Experiencer is an active participant in the Event, and in such cases the Event itself is often not mentioned, but rather some Focal_participant which is subordinately involved.
Core:	Event, Experiencer, Focal_participant, Location_of_Event
Non-core:	Cause, Degree, Duration, Manner, Place, Purpose_of_Event, Reason, Role_of_focal_participant, Time, Time_of_Event

BFN frames and FEs

want.v..6412

känna_för.vb..1

Examples	Valence patterns	
40	Event	Experiencer
(22)	VPto.Dep	NP.Ext
14	Experiencer	Focal_participant
(10)	NP.Ext	NP.Obj
(1)	PP[by].Dep	NP.Ext

Some valence patterns found in **BFN**

Examples	Valence patterns	
1	Event	Experiencer
(1)	VB.INF.VG	NN.SS
2	Experiencer	Focal_participant
(2)	NN.SS	NN.OO

Some valence patterns found in **SweFN**

FrameNet-based grammar in GF

- Existing FNs are not entirely formal and computational
  - We provide a **computational** FrameNet-based grammar and lexicon
- GF, Grammatical Framework (Ranta, 2004)
  - Separates between an **abstract syntax** and **concrete syntaxes**
  - Provides a general-purpose **resource grammar library** (RGL) for nearly 30 languages that implement the same abstract syntax
    - Large mono- and multilingual **lexicons** (for an increasing number of languages)
- The language-independent layer of FrameNet (frames and FEs) – the abstract syntax
  - The language-specific layers (surface realization of frames and LUs) – concrete syntaxes
- RGL is used for unifying the syntactic types used in different FNs
  - FrameNet allows for abstracting over RGL constructors

Specific aim (1)

- Provide a shared FrameNet API to GF RGL, so that application grammar developers could primarily use semantic constructors
  - In combination with some simple syntactic constructors
  - But instead of comparatively complex constructors for building verb phrases

```
mkCl person (mkVP (mkVP live_V) (mkAdv in_Prep place))
    -- mkCl : NP -> VP -> Cl
    -- mkVP : V -> VP
    -- mkVP : VP -> Adv -> VP
    -- mkAdv : Prep -> NP -> Adv
```

```
Residence                       -- Residence : NP -> Adv -> V -> Cl
    person                      -- NP (Resident)
    (mkAdv in_Prep place)       -- Adv (Location)
    live_V_Residence            -- V (LU)
```

Specific aim (2)

- FrameNet-annotated DBs of facts → multilingual CNL verbalization

		Time	Place	Relatives	**Child**	
Being_born	dzimt.v	1933. gada 3. maijs	Slokas pagasts	zvejnieka ģimene	Imants Ziedonis	
		Institution	**Subject**	Time	Place	**Student**
Education_teaching	absolvēt.v	Tukuma 1. vidusskola		1952. gads	Tukums	Imants Ziedonis
Education_teaching	beigt.v	Latvijas Universitāte	vēsture un filoloģija	1959. gads		Imants Ziedonis
Education_teaching	beigt.v	Augstākais literārais [..]		1964. gads	Maskava	Imants Ziedonis
		Employer	**Place_of_employment**	**Position**	Time	**Employee**
Being_employed	redaktors.n	izdevniecība Liesma		> redaktors		Imants Ziedonis
Being_employed	sekretārs.n	Latvijas rakstnieku [..]		> sekretārs		Imants Ziedonis
Being_employed	loceklis.n	AP tautas izglītības [..]		> loceklis		Imants Ziedonis
Being_employed	loceklis.n	Latvijas Institūts		> loceklis	1998. gads	Imants Ziedonis
Being_employed	padomnieks.n			> padomnieks	1997. gads	Imants Ziedonis
Being_employed	skolotājs.n	Jūrmalas 1. vidusskola		> skolotājs		Imants Ziedonis
		Time	**Prize**	Rank	Competition	**Competitor**
Win_prize	apbalvot.v	1983. gads	Tautu draudzības [..]			Imants Ziedonis
Win_prize	piešķirt.v	1972. gads	Nopelniem bagātais [..]			Imants Ziedonis
Win_prize	piešķirt.v	1977. gads	Tautas dzejnieka goda [..]			Imants Ziedonis
Win_prize	saņemt.v		1991. gada barikāžu [..]			Imants Ziedonis

- Issues
  - LU: a verb (which one?) or a copula (i.e., no LU)?
  - Prepositional object / adverbial modifier: which preposition (or case)?
  - Translation of FE fillers

Extraction of frame valence patterns

- Valence patterns that are shared between FNs (currently, BFN and SweFN)
  - Multilingual applications
  - Cross-lingual validation
- Currently, only core FEs that make the frames unique
- Example: the shared patterns for the frame *Desiring*
  - **Desiring**/V_{Act} Experiencer/NP_{Subj} Focal_participant/Adv
    e.g., [*Dexter*]$_{Experiencer}$ [*YEARNED*] [*for a cigarette*]$_{Focal\_participant}$
  - **Desiring**/$V2_{Act}$ Experiencer/NP_{Subj} Focal_participant/NP_{DObj}
    e.g., [*she*]$_{Experiencer}$ [*WANTS*] [*a protector*]$_{Focal\_participant}$
  - **Desiring**/VV_{Act} Event/VP Experiencer/NP_{Subj}
    e.g., [*I*]$_{Experiencer}$ *would n't* [*WANT*] [*to know*]$_{Event}$
- The uniform patterns contain sufficient info for generating the grammar

1. Language- and FN-specific processing

```xml
<sentence ID="732945">
 <text>Traders in the city want a change.</text>
 <annotationSet><layer rank="1" name="BNC">
  <label start="0" end="6" name="NP0"/>
  <label start="20" end="23" name="VVB"/>
  <label start="25" end="25" name="AT0"/>
 </layer></annotationSet>
 <annotationSet status="MANUAL">
  <layer rank="1" name="FE">
   <label start="0" end="18" name="Experiencer"/>
   <label start="25" end="32" name="Event"/>
  </layer>
  <layer rank="1" name="GF">
   <label start="0" end="18" name="Ext"/>
   <label start="25" end="32" name="Obj"/>
  </layer>
  <layer rank="1" name="PT">
   <label start="0" end="18" name="NP"/>
   <label start="25" end="32" name="NP"/>
  </layer>
  <layer rank="1" name="Target">
   <label start="20" end="23" name="Target"/>
  </layer>
 </annotationSet>
</sentence>
```

```xml
<sentence id="ebca5af9-e0494c4e">
 ...
 <w pos="VB" ref="3" deprel="ROOT">skulle</w>
 <element name="Experiencer">
  <w pos="PN" ref="4" dephead="3" deprel="SS">
   jag
  </w>
 </element>
 <element name="LU">
  <w msd="VB.AKT" ref="5" dephead="3" deprel="VG">
   vilja
  </w>
 </element>
 <element name="Event">
  <w msd="VB.INF" ref="6" dephead="5" deprel="VG">
   ha
  </w>
  <w pos="RG" ref="7" dephead="8" deprel="DT">
   sju
  </w>
  <w pos="NN" ref="8" dephead="6" deprel="OO">
   sångare
  </w>
 </element>
</sentence>
```

- Different XML schemes, POS tagsets and syntactic annotations
- Rules and heuristics for generalizing to RGL types, and for deciding the syntactic roles
- A lot of automatic annotation errors → heuristic correction (partial)

2. Extracted sentence patterns (BFN)

Desiring	Act	Experiencer_NP.*Subj* Event_VP	long.v
Desiring	Act	Experiencer_NP.*Subj* Event_VP Opt_Reason_Adv	aspire.v
Desiring	Act	Experiencer_NP.*Subj* Opt_Time_Adv Event_VP	fancy.v
Desiring	Act	Experiencer_NP.*Subj* Event_VP	want.v
Desiring	Act	Experiencer_NP.*Subj* Event_VP	yearn.v
Desiring	Act	Experiencer_NP.*Subj* Experiencer_NP.*Subj* Event_VP	aspire.v
Desiring	Act	Experiencer_NP.*Subj* Event_NP.*DObj*	want.v
Desiring	Act	Experiencer_NP.*Subj* Event_S	desire.v
Desiring	Act	Experiencer_NP.*Subj* Focal_participant_Adv[*after*]	yearn.v
Desiring	Act	Experiencer_NP.*Subj* Focal_participant_Adv[*for*]	yearn.v
Desiring	Act	Experiencer_NP.*Subj* Focal_participant_Adv[*for*]	yearn.v
Desiring	Act	Experiencer_NP.*Subj* Focal_participant_Adv	want.v
Desiring	Act	Experiencer_NP.*Subj* Focal_participant_NP.*DObj*	want.v
Desiring	Act	Experiencer_NP.*Subj* Focal_participant_NP.*DObj*	want.v
Desiring	Act	Focal_participant_NP.*DObj* Experiencer_NP.*Subj*	crave.v
Desiring	Act	Focal_participant_NP.*DObj*	want.v
Desiring	Pass	Focal_participant_NP.*Subj* Experiencer_NP.*DObj*	desire.v
Desiring	Pass	Focal_participant_NP.*Subj* Experiencer_NP.*DObj*	want.v

3. Summarized valence patterns (BFN)

```
Desiring : 288
    Act : 275
        Event_VP Experiencer_NP : 61
            Experiencer_NP.Subj Event_VP : 59
            Event_VP Experiencer_NP.Subj : 2
        Experiencer_NP Focal_participant_NP : 61
            Experiencer_NP.Subj Focal_participant_NP.DObj : 55
            Focal_participant_NP.DObj Experiencer_NP.Subj : 6
        Experiencer_NP Focal_participant_Adv : 43
            Experiencer_NP.Subj Focal_participant_Adv[for] : 26
            Experiencer_NP.Subj Focal_participant_Adv[after] : 7
            Experiencer_NP.Subj Focal_participant_Adv : 2
            ...
        ...
    Pass : 13
        Experiencer_NP Focal_participant_NP : 5
            Focal_participant_NP.Subj Experiencer_NP.DObj : 5
        ...
```

- Normalized, ignoring the word order and prepositions (or cases)
- For the abstract syntax, we consider only the normalized patterns
- For the concrete syntax – the most frequent sentence pattern of each normalized pattern

4. Pattern comparison by subsumption

- Pattern *A* subsumes pattern *B* if:
  - A.*frame* = B.*frame*
  - *type*(A.*LU*) = *type*(B.*LU*)
  - A.*voice* = B.*voice*
  - B.*FEs* ⊆ A.*FEs* (incl. the syntactic types and roles)
- If *A* subsumes *B* and *B* subsumes *A* then *A* = *B*
- If a pattern of FN_1 is subsumed by a pattern of FN_2, it is added to the **shared** set (and vice versa)
  - In the final set, patterns that are subsumed by other patterns are removed

```
P1: Apply_heat  V2  Act  Cook_NP.Subj Food_NP.DObj
P2: Apply_heat  V2  Act  Cook_NP.Subj Container_Adv Food_NP.DObj
P3: Apply_heat  V2  Act  Food_NP.DObj
```

P1 is subsumed by P2, P3 is subsumed by P1, P2; P1 and P3 are to be removed

Experiment series

- To roughly estimate the impact of various choices made in the extraction process, we have run a series of experiments

Settings	BFN							SweFN						
	Frames	Valence patterns		Sentence patterns		Corpus examples		Frames	Valence patterns		Sentence patterns		Corpus examples	
		total	per frame	total	per valence pattern	total	per sentence pattern		total	per frame	total	per valence pattern	total	per sentence pattern
0.0	556	19905	36	25696	1.3	68653	2.7	637	3395	5	3426	1.0	3688	1.1
2.0	555	16479	30	24491	1.5	66322	2.7	631	2811	4	2912	1.0	3393	1.2
2.A	551	14135	26	22115	1.6	65008	2.9	625	2704	4	2812	1.0	3275	1.2
2.B	551	5493	10	8696	1.6	65103	7.5	**625**	**1965**	**3**	**2089**	**1.1**	**3298**	**1.6**
3.0	506	6381	13	14393	2.3	56224	3.9	273	392	1	493	1.3	974	2.0
3.A	502	5968	12	13948	2.3	56841	4.1	266	381	1	489	1.3	952	1.9
3.B	**508**	**3481**	**7**	**6684**	**1.9**	**63091**	**9.4**	423	630	1	754	1.2	1963	2.6

0.0: Extract sentence patterns using FN-specific syntactic types ("baseline")
1.0: Skip examples containing few currently unconsidered syntactic types
2.0: Generalize syntactic types according to RGL
3.0: Skip once-used valence patterns (e.g., to reduce the propagation of annotation errors)

x.A: Skip repeated FEs
***x*.B**: Skip non-core FEs and repeated FEs

- In the result, we have extracted a set of **714** shared semantico-syntactic valence patterns covering **421** frames

FrameNet-based grammar: abstract

- **Frame** valence patterns are represented by functions
  - Taking one or more core FEs and one LU as arguments
  - Returning an object of type *Clause* whose linearization type is `{np: NP; vp: VP}`

```
fun Desiring_V       : Experiencer_NP -> Focal_participant_Adv -> V -> Clause
fun Desiring_V2      : Experiencer_NP -> Focal_participant_NP -> V2 -> Clause
fun Desiring_V2_Pass : Experiencer_NP -> Focal_participant_NP -> V2 -> Clause
fun Desiring_VV      : Event_VP -> Experiencer_NP -> VV -> Clause
```

- **FE**s are declared as semantic categories subcategorized by RGL types
  - *NP*, *VP*, *Adv* (includes prepositional phrases), *S* (embedded sentences)

```
cat Event_VP                          cat Focal_participant_NP
cat Experiencer_NP                    cat Focal_participant_Adv
```

- **LU**s are represented as functions that take no arguments
  - Return *V*, *V2*, *V3*, *VV*, *VS*, *V2V*, or *V2S*

```
fun hunger_V_Desiring : V             fun längta_V_Desiring : V
fun yearn_V_Desiring  : V             fun känna_V2_Desiring : V2
fun want_V2_Desiring  : V2            fun känna_VV_Desiring : VV
fun want_VV_Desiring  : VV            fun vilja_VV_Desiring : VV
fun yearn_VV_Desiring : VV            fun känna_V_Feeling : V
                                      fun känna_V2_Familiarity : V2
```

FrameNet-based grammar: concrete

- The mapping from the semantic FrameNet types to the syntactic RGL **types** is shared for all languages

```
lincat Focal_participant_NP  = Maybe NP
lincat Focal_participant_Adv = Maybe Adv
```

  – Linearization types are of type *Maybe* to allow for optional (empty) FEs

- To implement the frame functions, RGL **constructors** are applied to the arguments depending on their types and syntactic roles, and the voice

```
lin Desiring_V2 experiencer focal_participant v2 = {
  np = fromMaybe NP experiencer ;
  vp = mkVP v2 (fromMaybe NP focal_participant)
}

lin Desiring_V2_Pass experiencer focal_participant v2 = {
  np = fromMaybe NP focal_participant ;
  vp = mkVP (passiveVP v2) (mkAdv by8agent_Prep (fromMaybe NP experiencer))
}
```

- The monolingual RGL **dictionaries** are reused for implementing LUs
  – 2,755 (2,996) entries for English, and 1,211 (1,257) for Swedish

FrameNet-based grammar: concrete

Verb	Voice	FE types and roles	Freq.	Verb	Voice	FE types and roles	Freq.
V2	Act	NP_{DObj} NP_{Subj}	238	V	Act	Adv	8
V	Act	Adv NP_{Subj}	138	V2	Act	Adv NP_{DObj}	8
V2	Pass	NP_{Subj}	70	V2V	Act	NP_{IObj} NP_{Subj} VP	5
V	Act	NP_{Subj}	65	VS	Pass	S	3
V2	Act	Adv NP_{DObj} NP_{Subj}	62	V	Act	Adv Adv Adv NP_{Subj}	2
V2	Pass	Adv NP_{Subj}	31	V2	Act	Adv Adv NP_{DObj} NP_{Subj}	2
VS	Act	NP_{Subj} S	26	V2	Pass	Adv	2
VV	Act	NP_{Subj} VP	18	V2	Pass	Adv Adv NP_{Subj}	2
V	Act	Adv Adv NP_{Subj}	14	V3	Act	NP_{IObj} NP_{Subj}	2
V2	Act	NP_{DObj}	14	VS	Act	Adv NP_{Subj} S	2

- The 714 semantico-syntactic valence patterns reuse **25 syntactic patterns**
  - 25 **RGL-based code templates** are used to generate the implementation of frame functions; most templates are derived from few basic templates
    - E.g., adverbial modifiers are added by recursive calls of the *mkVP* constructor (the order of *Adv* FEs can differ across languages)

A FrameNet-based API to GF Resource Grammar Library

Frames | Verbs

vers. 0.9.3

REMU

Damaging
Daring
Death
Deciding
Delimitation_of_diversity
Delivery
Deny_permission
Departing
Deserving
Desiring
Destroying
Detaching
Detaining
Differentiation
Dispersal
Dodging
Dominate_competitor
Dominate_situation
Dressing
Drop_in_on
Dunking
Earnings_and_losses
Eclipse
Education_teaching
Elusive_goal
Emanating
Emitting
Emotion_active
Emotion_heat
Employing
Emptying
Encoding
Endangering
Enforcing
Entering_of_plea
Escaping

Desiring

Desiring_V : Experiencer_NP → Focal_participant_Adv → V → Clause

- ▶ **Eng**: [he]Experiencer [WANTED] [more]Focal_participant
  - ▶ **aspire_V_Desiring** : V
  - ▶ **hanker_V_Desiring** : V
  - ▶ **hunger_V_Desiring** : V
  - ▶ **long_V_Desiring** : V
  - ▶ **lust_V_Desiring** : V
  - ▶ **pine_V_Desiring** : V
  - ▶ **thirst_V_Desiring** : V
  - ▶ **want_V_Desiring** : V
  - ▶ **yearn_V_Desiring** : V
- ▶ **Swe**: [Roberte]Experiencer [LÄNGTADE] [hem till Tyskland]Focal_participant
  - ▶ **längta_V_Desiring** : V

Desiring_V2 : Experiencer_NP → Focal_participant_NP → V2 → Clause

- ▶ **Eng**: [you]Experiencer [WANT] [one]Focal_participant
  - ▶ **covet_V2_Desiring** : V2
  - ▶ **crave_V2_Desiring** : V2
  - ▶ **desire_V2_Desiring** : V2
  - ▶ **fancy_V2_Desiring** : V2
  - ▶ **want_V2_Desiring** : V2
  - ▶ **yearn_V2_Desiring** : V2

http://grammaticalframework.org/framenet/

Case study: Phrasebook

- Precise translation of standard touristic phrases
- Apart from idiomatic phrases, many can be constructed by aplying the previously introduced frame functions

- `ALive : Person -> Country -> Action`
  - `Residence_V : Location_Adv -> Resident_NP -> V -> Clause`
    - *I live in Sweden* (Eng)
    - *jag bor i Sverige* (Swe)

- `AWantGo : Person -> Place -> Action`
  - `Desiring_VV : Event_VP -> Experiencer_NP -> VV -> Clause`
  - `Motion_V_2 : Goal_Adv -> Source_Adv -> Theme_NP -> V -> Clause`
    - *we want to go to a museum* (Eng)
    - *vi vill gå till ett museum* (Swe)

- No changes needed in the Phrasebook abstract syntax
  - Frame functions are not part of Phrasebook abstract syntax trees
- The re-engineered grammar generates equal phrases

Case study: Phrasebook

- **Before**:

```
lin ALive p co =
  mkCl
    p.name
    (mkVP
      (mkVP (mkV "live"))
      (mkAdv in_Prep co))
```

```
lin AWantGo p pl =
  mkCl
    p.name
    want_VV
    (mkVP
      (mkVP IrregEng.go_V)
      pl.to)
```

- **After**:

```
lin ALive p co = let cl : Clause =
    Residence_V
      (Just Adv (mkAdv in_Prep co))
      (Just NP p.name)
      live_V_Residence
    in mkCl cl.np cl.vp
```

```
lin AWantGo p pl = let cl : Clause =
    Desiring_VV
      (Just VP                    -- Event
        (Motion_V_2
          (Just Adv pl.to)        -- Goal
          (Nothing' Adv)          -- Source
          (Nothing' NP)           -- Theme
          go_V_Motion
        ).vp)
      (Just NP p.name)            -- Experiencer
      want_VV_Desiring
    in mkCl cl.np cl.vp
```

Case study: Painting grammar

- Verbalizes descriptions of museum objects stored in an ontology
- A set of triples describing the artwork *Le Général Bonaparte*:
  - `<LeGeneralBonaparte> <createdBy> <JacquesLouisDavid>`
  - `<LeGeneralBonaparte> <hasDimension> <LeGeneralBonaparteDimesion>`
  - `<LeGeneralBonaparte> <hasCreationDate> <LeGeneralBonaparteCreationDate>`
  - `<LeGeneralBonaparte> <hasCurrentLocation> <MuseeDuLouvre>`
- Triples are combined by the grammar to generate a <u>coherent text</u>
  - `DPainting : Painting -> Painter -> Year -> Size -> Museum -> Description`
    - Eng: *Le Général Bonaparte* ***was painted by*** *Jacques-Louis David* ***in*** *1510.* *<u>It</u>* ***measures*** *81 by 65 cm.* *<u>This work</u>* ***is displayed at*** *the Musée du Louvre.*
    - Swe: *Le Général Bonaparte* ***målades av*** *Jacques-Louis David* ***år*** *1510.* *<u>Den</u>* ***mäter*** *81 gånger 65 cm.* *<u>Det här verket</u>* ***hänger på*** *Louvren.*
- The re-engineered grammar generates **semantically equivalent** descriptions
  - The Swedish grammar uses different verbs and pronouns in comparison to English and the original Swedish grammar

Case study: Painting grammar

```
lin DPainting
 painting painter year size museum =
let
 s1 : Text = mkText (mkS
  pastTense (mkCl painting (mkVP
   (mkVP (passiveVP paint_V2)
    (mkAdv by8agent_Prep
     painter.long)) year.s))) ;

 s2 : Text = mkText
  (mkCl it_NP (mkVP (mkVP
   (mkVPSlash measure_V2)
   (mkNP (mkN "")) size.s))) ;

 s3 : Text = mkText
  (mkCl (mkNP this_Det painting)
   (mkVP (passiveVP display_V2)
    museum.s))

in mkText s1 (mkText s2 s3) ;
```

```
lin DPainting
 painting painter year size museum =
let
 cl1 : Clause =
  Create_physical_artwork_V2_Pass*
   (Just NP painter.long) -- Creator
   (Just NP painting)     -- Representation
   paint_V2_Create_physical_artwork ;

 cl2 : Clause = Dimension_V2*
  (Just NP size.s) -- Measurement
  (Just NP it_NP)  -- Object
  measure_V2 ;

 cl3 : Clause = Being_located_V2_Pass*
  (Just Adv museum.s)                   -- Loc.
  (Just NP (mkNP this_Det painting)) -- Theme
  display_V2

in mkText (mkText (mkS pastTense
 (mkCl cl1.np (mkVP cl1.vp year.s))) -- Time
 (mkText (mkCl cl2.np cl2.vp)
  (mkText (mkCl cl1.np cl3.vp))) ;
```

* Currently not available out-of-the-box

Evaluation

- **Intrinsic**
  - The number of examples in the source corpora that belong to the set of shared frames..
    - ..and are covered by the shared semantico-syntactic valence patterns
  - Corpus examples are represented by sentence patterns disregarding non-core FEs, word order and prepositions
    - Syntactic roles and the grammatical voice are considered
  - In **BFN**, ~55,800 examples (**84.1%** of total) belong to the shared set of 421 frames, and **69.4%** of them are covered by the shared patterns
    - In **SweFN**, ~2,400 examples (**71.4%** of total) belong to the shared set of frames, and **69.0%** of them are covered by the shared patterns
- **Extrinsic**
  - The number of constructors used to linearize functions in the original vs. re-engineered grammar (comparison of code complexity)
    - In Paintings, the number of constructors is reduced by 38% while in Phrasebook – by 20–27% (considering only the modified functions)

Summary

- A novel approach for automatic acquisition of a multilingual semantic grammar from FrameNet-annotated corpora
  - A **unified** method to compare semantico-syntactic valence patterns across FNs
- Despite the small SweFN corpus, the set of extracted shared valence patterns is **concise** and already provides a wide coverage
  - The relatively small number of patterns allows for manual checking
  - The numbers are not stable and vary across releases but illustrate the tendency
- The FrameNet API to RGL makes certain application grammars more robust and flexible (easier to extend)
- The valence extracted for LUs provides feedback to RGL dictionaries
- The future potential is to provide a means for multilingual verbalization of FrameNet-annotated databases

Future work

- Add more languages
  - Treebank-based corpora (e.g. German)
  - Rich morphology (e.g. Latvian)
- Detect **prepositional objects** (*NP* vs. *Adv*; LU-governed prepositions)
- Differentiate syntactic **roles of *VP* FEs** (object vs. adverbial modifier)
- Include shared **non-core FEs** (via a modified comparison algorithm)
- **Align LUs** among languages (e.g. via GF translation dictionaries)
- Towards FrameNet parsing in GF
  - First, frame labelling
    - FrameNet grammar as an **embedded CNL** in RGL
    - Restrict **LUs to frames** (by using GF dependent types)
  - Later, full semantic role labelling (SRL)