Querying a Dozen Corpora and a Thousand Years with Fintan

Prêt-à-LLOD

Christian Chiarcos[+*]
chiarcos@em.uni-frankfurt.de

Christian Fäth[+]
faeth@informatik.uni-frankfurt.de

Maxim Ionov[+*]
ionov@cs.uni-frankfurt.de

[+] Applied Computational Linguistics (ACoLi), Goethe-Universität Frankfurt, Germany
[*] Institute for Digital Humanities (IDH), Universität zu Köln, Germany

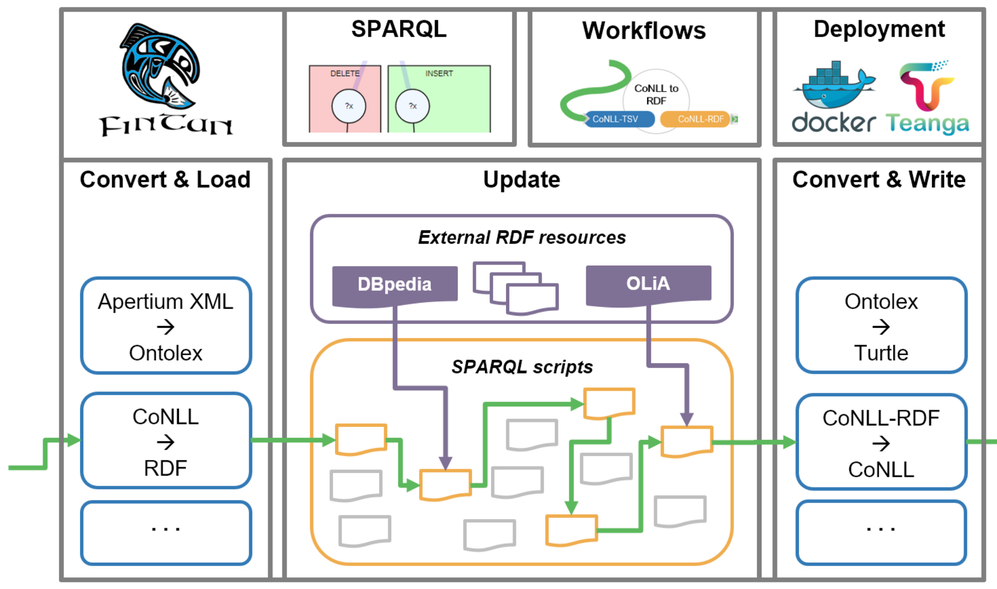
Fintan – Flexible INtegrated Transformation and Annotation eNgineering

Transformating heterogeneous data in a unified way
(Fäth et al. 2020)

- Convert *any* kind of language resource to RDF graphs.
- Manipulate/link/transform graphs with SPARQL.
- Serialize as RDF or in conventional corpus formats

Modular: Pipelines broken into small, reusable pieces
Reusable: Same RDF vocabulary => same modules
Extensible: Add your own (SPARQL, Docker, Java, ...)
Scalable: Stream processing & parallelization

https://github.com/Pret-a-LLOD/Fintan (wrapper repo)
https://github.com/acoli-repo/conll-rdf (CoNLL customization)



So far, used for transformation and annotation engineering.
Here: Querying & Retrieval !

Use Case: Order of Nominal Dative and Accusative Arguments in Historical German

Polomeus schreib die tat dem kunig
Ptolemeus write.PST the.**F.Acc** deed.**F.Acc** the.**M.Dat** king.**M.Dat**
(*Ptolemeus wrote the deed to the king.*)
[Middle High German]

Si zeig dem kunig den mandel und ir gewant
She show.PST the.**M.Dat** king.**M.Dat** the.**M.Acc** coat.**M.Acc** and **her.Acc** garment.**N.Acc**
(*She showed her coat and garment to the king*) **[Middle High German]**

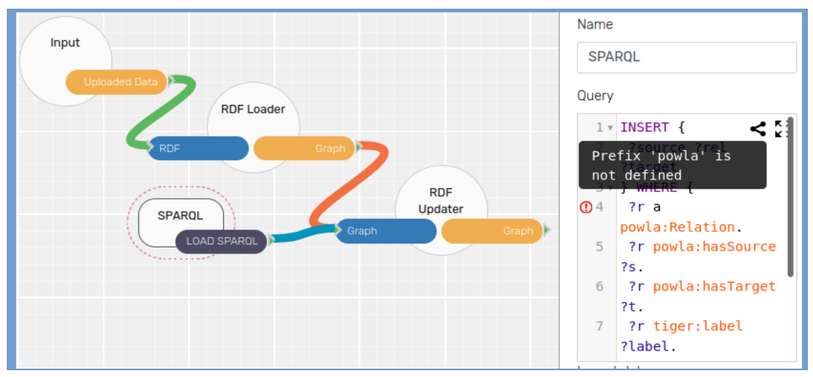
Fintan Workflow

Preprocessing => input format(s) (outside Fintan)

Fintan Loader => RDF (for XML, CoNLL, ...)
Fintan Updater => uniform data model
Fintan Updater => query pre-compilation
(optional, e.g., labelled edges => object properties)
Fintan Formatter export and/or query

Evaluation => external tools

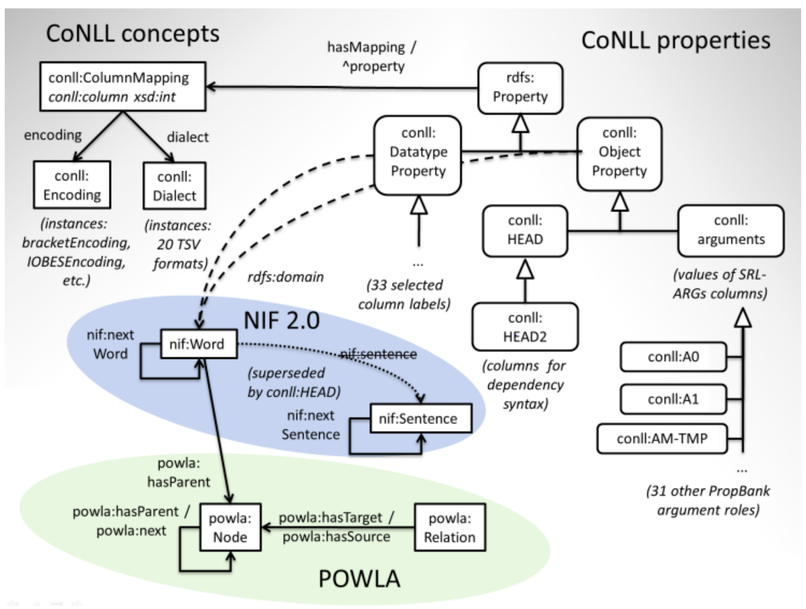
Fintan Workflow Editor



Uniform Data Model

RDF vocabularies:
- nif (words, sentences)
- conll (dependencies, word annotations)
- powla (phrases, trees, labelled edges)
- tiger* (phrase and edge annotations)

* currently points to TIGER-XML documentation



12 corpora, 1000 years => heterogeneous:
- dependency syntax (4 formats)
- phrase structure syntax (3 formats)
- flat span annotations (1 format)

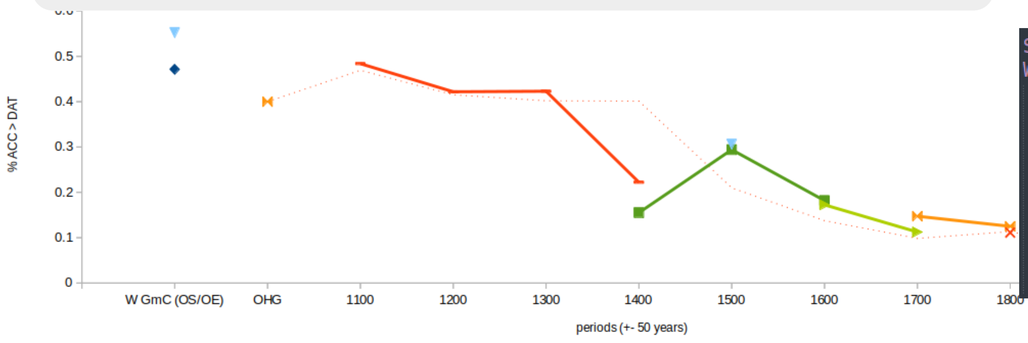
Observations

- Fintan provides natural support for querying with SPARQL
- Queries wrapped in Fintan workflowcan be easily replicated
- Unlike query tools like ANNIS, SPARQL supports deduplication in aggregates

We explore routes to integration

Results

- Smoothed over 100 year windows, the corpus statistics form a continuous plot
- Continuous trend to restrict ACC>DAT since the 14th c.



Example (Spans)

(Old High German TCodex, visualized with ANNIS)

Inti her ... / furliez sîna quenun sinemo bruoder

default_ns (grid)

LAT	&		...	/	reliquit	uxorem	suam	fratri	suo
bibl	T 208, 4-5 Alpha´ Mt 22								
cat		NP			VP	NP		NP	
clause-status		MAINDECL							
comment	Subjektspronomen im Vorfeld eingefügt								
context	AR								
definiteness		DEF				DEF		DEF	
foc-bg								NIF	
gf		SUBJ			VFIN	DO		IO	
givenness		GIV				GIV		ACC	
pos	CONJ	PRONPRS			V	PRONPOS	N	PRONPOS	N
position		GV							
syl_no	2	1			2	4		5	
tok	Inti	her	...	/	furliez	sîna	quenun	sinemo	bruoder

Preprocessing (CoNLL)

(CoNLL conversion, cf. Chiarcos & Schenk 2018)

# ID	LAT	words	align	pos	cat	clause-status	gf	...
T0	&	Inti	_	CONJ		_	_	
T6	_	her	_	PRONPRS	NP	B-MAINDECL	SUBJ	
T1	...	...	_	_	_	I-MAINDECL	_	
T2	/	/	_	_	_	I-MAINDECL	_	
T5	reliquit	furliez	_	V	VP	I-MAINDECL	VFIN	
T9	uxorem	sîna	_	PRONPOS	B-NP	I-MAINDECL	B-DO	
T7	suam	quenun	_	N	E-NP	I-MAINDECL	E-DO	
T8	fratri	sinemo	_	PRONPOS	B-NP	I-MAINDECL	B-IO	
T3	suo	bruoder	_	N	E-NP	E-MAINDECL	E-IO	

Loader/Updater (CoNLL-RDF)

IOB-notation is resolved during loading ... but nesting of spans is to be asserted with SPARQL

```
DELETE { ?w powla:hasParent ?gf, ?cl }
INSERT { ?phrase powla:hasParent ?gf.
         ?gf powla:hasParent ?gl }
WHERE { ?w powla:hasParent ?phrase, ?gf, ?cl.
        ?phrase a conll:CAT. # one column
        ?gf a conll:GF.      # per original
        ?cl a conll:CLAUSE.  # tier
};
```

Querying (SPARQL)

```
SELECT DISTINCT ?do ?io ?order
WHERE {
  [ tiger:label "DO" ] powla:hasSource ?cl ;
                       powla:hasTarget ?do .
  [ tiger:label "IO" ] powla:hasSource ?cl ;
                       powla:hasTarget ?io .
  ?acc powla:hasParent* ?do.
  ?dat powla:hasParent* ?io.
  OPTIONAL {
    ?acc nif:nextWord+ ?dat. BIND("DAT>ACC" as ?order )
  } }
```