

Federated queries over heterogeneous IGT collections

Maxim Ionov
mionov@uni-koeln.de

University of Cologne, Germany

Open Text Collections Workshop, 12.12.23

Outline

- 1 Background: IGT and Linked Data
- 2 Ligt vocabulary
- 3 Using Ligt

Outline

- 1 Background: IGT and Linked Data
- 2 Ligt vocabulary
- 3 Using Ligt

What is IGT

This page intentionally left blank.

Motivation

My fieldwork background fieldwork by Moscow State University

Motivation

My fieldwork background fieldwork by Moscow State University

- Data for a single language

Motivation

My fieldwork background fieldwork by Moscow State University

- Data for a single language
 - gathered by many people
 - over several years
 - in Word or plain text

Motivation

My fieldwork background fieldwork by Moscow State University

- Data for a single language
 - gathered by many people
 - over several years
 - in Word or plain text
- Extremely difficult to work with

Motivation

My fieldwork background fieldwork by Moscow State University

- Data for a single language
 - gathered by many people
 - over several years
 - in Word or plain text
- Extremely difficult to work with
 - reliably search across the data
 - cross-reference and check

Motivation

My fieldwork background fieldwork by Moscow State University

- Data for a single language
 - gathered by many people
 - over several years
 - in Word or plain text
- Extremely difficult to work with
 - reliably search across the data
 - cross-reference and check
- From 2006 to 2010 nominalisation strategy in Ossetian spoken in Dargavs changed according to fieldworkers' memory.

Motivation

My fieldwork background fieldwork by Moscow State University

- Data for a single language
 - gathered by many people
 - over several years
 - in Word or plain text
- Extremely difficult to work with
 - reliably search across the data
 - cross-reference and check
- From 2006 to 2010 nominalisation strategy in Ossetian spoken in Dargavs changed according to fieldworkers' memory.
- Although it was impossible to check the data and make a reliable conclusion.

Motivation

Additional motivation:

Motivation

Additional motivation:

- Data presented during talks and in the papers might look convincing
 - especially if the language / phenomenon is not your primary expertise
 - or there are a lot of examples and resercher talks/writes with confidence

Motivation

Additional motivation:

- Data presented during talks and in the papers might look convincing
 - especially if the language / phenomenon is not your primary expertise
 - or there are a lot of examples and resercher talks/writes with confidence
- Not all the data was gathered well

Motivation

Additional motivation:

- Data presented during talks and in the papers might look convincing
 - especially if the language / phenomenon is not your primary expertise
 - or there are a lot of examples and resercher talks/writes with confidence
- Not all the data was gathered well
- Careful checking require going through a lot of data, often in different formats, etc.

Idea and requirements

- An interface allowing searching across different datasets at the same time

Idea and requirements

- An interface allowing searching across different datasets at the same time
- Supporting multiple input formats

Idea and requirements

- An interface allowing searching across different datasets at the same time
- Supporting multiple input formats
- Basic search

Idea and requirements

- An interface allowing searching across different datasets at the same time
- Supporting multiple input formats
- Basic search
- Filtering

Idea and requirements

- An interface allowing searching across different datasets at the same time
- Supporting multiple input formats
- Basic search
- Filtering
- Export

Idea and requirements

- An interface allowing searching across different datasets at the same time
- Supporting multiple input formats
- Basic search
- Filtering
- Export

gathering → filtering → using

Challenges (IGT as data)

- No fixed set of layers
- Alternative analyses
- Multiple different (partly incompatible) formats

⇒ Leads to separate silos of data

Yet another format? C'mon!

- Adopting a format does not always depend on how good the format is

Yet another format? C'mon!

- Adopting a format does not always depend on how good the format is
 - Habit
 - Courses
 - Experience and support
 - Interfaces

Yet another format? C'mon!

- Adopting a format does not always depend on how good the format is
 - Habit
 - Courses
 - Experience and support
 - Interfaces
- Instead of setting up to create a full solution

Yet another format? C'mon!

- Adopting a format does not always depend on how good the format is
 - Habit
 - Courses
 - Experience and support
 - Interfaces
- Instead of setting up to create a full solution
 - Focus on interoperability and compatibility with other formats

Yet another format? C'mon!

- Adopting a format does not always depend on how good the format is
 - Habit
 - Courses
 - Experience and support
 - Interfaces
- Instead of setting up to create a full solution
 - Focus on interoperability and compatibility with other formats
 - And on support of off-the-shelf tools

In sum

TLDR: I want a tool that takes different collections of IGT prepared in other programs in different formats and allows to search in them simultaneously

In sum

TLDR: I want a tool that takes different collections of IGT prepared in other programs in different formats and allows to search in them simultaneously

There is a technology that provides most of the requirements out of the box

Linked Data

Linked Data was created as an extension of the WWW principles to the real-world objects

Linked Data

Linked Data was created as an extension of the WWW principles to the real-world objects

- ① Uniform Resource Identifiers (URIs) should be used to name and identify individual things.
- ② HTTP URIs should be used to allow these things to be looked up, interpreted, and subsequently "dereferenced".
- ③ Useful information about what a name identifies should be provided through open standards such as RDF, SPARQL, etc.
- ④ When publishing data on the Web, other things should be referred to using their HTTP URI-based names.

Linked Data

Linked Data was created as an extension of the WWW principles to the real-world objects

- ① Uniform Resource Identifiers (URIs) should be used to name and identify individual things.
- ② HTTP URIs should be used to allow these things to be looked up, interpreted, and subsequently "dereferenced".
- ③ Useful information about what a name identifies should be provided through open standards such as RDF, SPARQL, etc.
- ④ When publishing data on the Web, other things should be referred to using their HTTP URI-based names.

Basically, it entails a **highly standartised text format(s)** with a tool stack

Linked Data

Linked Data was created as an extension of the WWW principles to the real-world objects

- ① Uniform Resource Identifiers (URIs) should be used to name and identify individual things.
- ② HTTP URIs should be used to allow these things to be looked up, interpreted, and subsequently "dereferenced".
- ③ Useful information about what a name identifies should be provided through open standards such as RDF, SPARQL, etc.
- ④ When publishing data on the Web, other things should be referred to using their HTTP URI-based names.

Basically, it entails a **highly standartised text format(s)** with a tool stack (but conceptually datasets are a multigraph)

Linked Data: Pros and cons

- + It is based on text files, so it does not depend on a platform / technological stack / etc.
- + Allows connecting datasets to each other, breaking silos and making data accessible
- + Promotes rigid vocabularies and standards making data interoperable (for the most part)
- + A lot of off-the-shelf tools allowing storing, searching, retrieving data

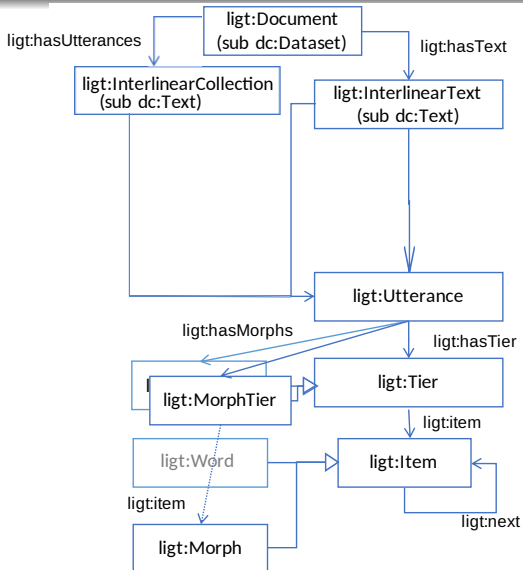
Linked Data: Pros and cons

- + It is based on text files, so it does not depend on a platform / technological stack / etc.
- + Allows connecting datasets to each other, breaking silos and making data accessible
- + Promotes rigid vocabularies and standards making data interoperable (for the most part)
- + A lot of off-the-shelf tools allowing storing, searching, retrieving data
 - Very steep learning curve, especially when things come to using the off-the-shelf tools
 - Rigid vocabularies often lead to hacky modelling which decreases interoperability
 - Slow for complex use-cases
 - Relatively obscure which means less support

Outline

- 1 Background: IGT and Linked Data
- 2 Ligt vocabulary
- 3 Using Ligt

Basic idea



Simple example

`https://s.zazuko.com/qD7XtT`

Simple example

```
PREFIX ligt: <http://purl.org/ligt/ligt-0.2#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT (COUNT(?lang) as ?n_lang) ?val
WHERE {
    ?morph ligt:gloss ?val ;
           rdfs:label ?label .

    BIND(LANG(?label) as ?lang)
    FILTER(?val = UCASE(?val) && ?lang != '')
} GROUP BY ?val ORDER BY DESC(?n_lang)
```

Marker	#
3SG	2650
1SG	2397
NEG	1400
2SG	1306
PST	1099

Outline

- 1 Background: IGT and Linked Data
- 2 Ligt vocabulary
- 3 Using Ligt

Using Ligt

- Conventional pipeline
 - Converting datasets
 - Setting up a SPARQL endpoint
 - Querying data

Using Ligt

- Conventional pipeline
 - Converting datasets
 - Setting up a SPARQL endpoint
 - Querying data
- **But:**
 - Strength of Linked Data in having many datasets provided by different people and organisations
 - Setting up the infrastructure is difficult and often there are no resources to sustain it
 - Even providing data dumps require people to convert their data to an unfamiliar format (and be aware of it)

Using Ligt

- Conventional pipeline
 - Converting datasets
 - Setting up a SPARQL endpoint
 - Querying data
- **But:**
 - Strength of Linked Data in having many datasets provided by different people and organisations
 - Setting up the infrastructure is difficult and often there are no resources to sustain it
 - Even providing data dumps require people to convert their data to an unfamiliar format (and be aware of it)

⇒ **on-the-fly conversion** via a service

On-the-fly conversion

- Data providers do not need to know about Ligt or put effort into creating and storing additional data (let alone setting up a SPARQL endpoint)

On-the-fly conversion

- Data providers do not need to know about Ligt or put effort into creating and storing additional data (let alone setting up a SPARQL endpoint)
- Ligt users do not need to worry about licensing and storing the data since potentially it is possible not to store the converted version

On-the-fly conversion

- Data providers do not need to know about Ligt or put effort into creating and storing additional data (let alone setting up a SPARQL endpoint)
- Ligt users do not need to worry about licensing and storing the data since potentially it is possible not to store the converted version
- It can even be done on the frontend with JS, no server required

Example: querying examples in cldf datasets

- `examples.csv` usually have the examples available in the dataset
- *Grambank* does not have this table, but there are examples in the text:

Feature GB020: Are there definite or specific articles?



Patrons: [Jeremy Collins](#) and [Jay Lataatche](#)

Summary

An article is a marker that accompanies the noun and expresses notions such as (non-)specificity and (in)definiteness. It may be free, bound, or marked by suprasegmental markers such as tone. Articles are different from demonstratives in that they are specific articles, they form a natural continuum, making it hard to define discrete categories, but to qualify as an article it must be used to refer to a specific entity.

Procedure

1. Code 1 if there is a morpheme that can mark definiteness or specificity without also conveying a spatial deictic meaning.
2. Code 0 if the source does not mention a definite article and you cannot find one in examples or texts in an otherwise similar context.
3. Code ? if the grammar does not contain enough analysis to determine whether there is a definite article or not.
4. If you have coded 1 for GB020 and 0 for GB021 and GB022, please write a comment explaining the position of the article.
- 5.

Examples

Aiton (ISO 639-3: aio, Glottolog: aito1238)

Coded 1. The definite article is postnominal (Morey 2005:244-245).

```
no      nan      a māt      ne      wā
time    that  minister DEF  say
‘Then the minister said.’
```

Buwai (ISO 639-3: bhw, Glottolog: buwa1243)

Coded 1 (Viljoen 2013: 234–242).

```
Ra      rga      teked      anta      vway ?
nā      rga      tōkōd      antā      vājāj
REL     break  calabash DEF  who
‘Who broke the calabash?’
```

Example: extracting 1SG morphs for several unconverted datasets

```
PREFIX ligt: <http://purl.org/ligt/ligt-0.3#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT ?gram ?lang  
{  
  ?s a ligt:Morph ;  
    rdfs:label ?gram ;  
    ligt:gloss "1SG"@en .  
  
  BIND(LANG(?gram) AS ?lang)  
}  
LIMIT 10
```

```
./comunica-sparql-file \  
https://converter/cldf/https://github.com/cldf-datasets/apics/  
.../cldf/examples.csv \  
https://converter/cldf-md/https://github.com/cldf-datasets/grambank/  
.../cldf/features.csv \  
-f get-grams.rq
```


Example: extracting all possible causatives for each language

```
PREFIX ligt: <http://purl.org/ligt/ligt-0.3#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT (group_concat(?gram; separator=" | ") as ?grams) ?morph_val ?la
{
  ?doc a ligt:Document ;
       ligt:hasUtterances/ligt:utterance/ligt:hasMorphs/ligt:item ?s .
  ?s a ligt:Morph ;
     rdfs:label ?gram ;
     ligt:gloss ?morph_val .

  BIND(LANG(?gram) AS ?lang)
  FILTER(REGEX(?morph_val, "CAUS"))
} GROUP BY ?lang ?doc ?morph_val LIMIT 100

***

./comunica-sparql-file ./apics ./grambank.ttl -f get-caus.rq
```

LLOD advantages: Mapping annotations

- If we know the glosses used in our datasets, we can link our datasets with external ontologies

LLOD advantages: Mapping annotations

- If we know the glosses used in our datasets, we can link our datasets with external ontologies
- In practice, this means adding one line for each mapping:

```
<http://purl.org/olia/unimorph.owl#ABL> apics:hasValue "ABL"@en .  
<http://purl.org/olia/unimorph.owl#ABS> apics:hasValue "ABS"@en .  
<http://purl.org/olia/unimorph.owl#ACC> apics:hasValue "ACC"@en .  
<http://purl.org/olia/unimorph.owl#ACT> apics:hasValue "ACT"@en .  
<http://purl.org/olia/unimorph.owl#ADJ> apics:hasValue "ADJ"@en .
```

LLOD advantages: Mapping annotations

- If we know the glosses used in our datasets, we can link our datasets with external ontologies
- In practice, this means adding one line for each mapping:

```
<http://purl.org/olia/unimorph.owl#ABL> apics:hasValue "ABL"@en .  
<http://purl.org/olia/unimorph.owl#ABS> apics:hasValue "ABS"@en .  
<http://purl.org/olia/unimorph.owl#ACC> apics:hasValue "ACC"@en .  
<http://purl.org/olia/unimorph.owl#ACT> apics:hasValue "ACT"@en .  
<http://purl.org/olia/unimorph.owl#ADJ> apics:hasValue "ADJ"@en .
```
- After this process we can operate with concepts: case, gender, aspect instead of strings

LLOD advantages

- Linking external resources: linking a dictionary, corpus or other resources

LLOD advantages

- Linking external resources: linking a dictionary, corpus or other resources
- Adding intermediate annotations: annotate what you found to find it easier next time

LLOD advantages

- Linking external resources: linking a dictionary, corpus or other resources
- Adding intermediate annotations: annotate what you found to find it easier next time
- When dealing with remote data (or converted remote data), it is possible to add and save annotations **locally**, keeping some notes, alternative annotations, etc.

Where do we go from here: Interfaces

- There is a lot of potential, but one drawback is obvious: lack of user-friendly interfaces

Where do we go from here: Interfaces

- There is a lot of potential, but one drawback is obvious: lack of user-friendly interfaces
- This is ongoing work, but ideally in the future users will not need to know that they use Linked Data

Where do we go from here: Interfaces

- There is a lot of potential, but one drawback is obvious: lack of user-friendly interfaces
- This is ongoing work, but ideally in the future users will not need to know that they use Linked Data
- Queryiing with SPARQL can be very difficult, so it needs to be hidden using a query builder

Where do we go from here: Interfaces

- There is a lot of potential, but one drawback is obvious: lack of user-friendly interfaces
- This is ongoing work, but ideally in the future users will not need to know that they use Linked Data
- Queryiing with SPARQL can be very difficult, so it needs to be hidden using a query builder
- What could be potential queries that make linguistic sense?

- ? What could be potential queries that make linguistic sense?
- ? What potential linguistic use-cases this can be used for?

<https://github.com/max-ionov/ligt>