

Integrating NLP with Linked Data and RDF: the NIF format (hands on)

Ciro Baron Neto

Ph.D student at University of Leipzig

- Github NLP2RDF web page overview and NIF Online demos (Dashboard, Combinator...)
- Examples
 - Example 1: How to annotate string
 - using Snowball Steamer and OpenNLP
 - Example 2:
 - Query generated NIF data and Querying Brown Corpus

- <https://github.com/NLP2RDF/>
- </home/ciro/websites/github/github.com/NLP2RDF/index.html>

NIF 2.0 Dashboard

This dashboard reflects the actual state of resources of the NIF 2.0 project. Here you can also find documentation, corpora, ontologies, demos, wikis and specifications.

Online REST Webservices		
Service Name	URL	Status
OpenNLP	http://demo.nlp2rdf.org:9998/opennlp?f=text&i=This+is+a+test.&t=direct&modelFolder=model	Online!
Stanford	http://demo.nlp2rdf.org:9999/stanfordcorenlpn?f=text&i=This+is+a+test.&t=direct	Online!
Snowball	http://demo.nlp2rdf.org:9996/snowball?f=text&i=This+is+my+favorite+test.&t=direct	Online!
NIF Validator	http://demo.nlp2rdf.org:9990	Online!
NIF DBpedia Spotlight	http://demo.nlp2rdf.org:9995/spotlight?f=text&i=This+is+Germany.&t=direct&confidence=0.3	Online!
NIF Combinator	http://demo.nlp2rdf.org	Online!

NIF Corpus		
Corpus Name	URL	Size
The LOD Wikilinks corpus	http://wiki-link.nlp2rdf.org/	500M
News-100 corpus	https://raw.githubusercontent.com/AKSW/n3-collection/master/News-100.ttl	13k
RSS-500 corpus	https://raw.githubusercontent.com/AKSW/n3-collection/master/RSS-500.ttl	10k
Reuters-128 corpus	https://raw.githubusercontent.com/AKSW/n3-collection/master/Reuters-128.ttl	7k
Brown corpus	http://brown.nlp2rdf.org/	500k
Add Corpus		

Ontology		NIF Specifications	
NIF Validator TestCase Suite@en	Details	W3C Conversion of NIF to ITS 2.0@en	Details
STC - Simple Test Case Ontology@en	Details	W3C Conversion of ITS 2.0 to NIF@en	Details
RLOG - an RDF Logging Ontology@en	Details	NIF 2.0 Core Specification@en	Details

Wikis	
How to publish a txt corpora with NIF as Linked Data@en	Details

Poster	
Promotional poster for Meta Forum@en	Details

Software	
DBpedia Spotlight implementation of NIF@en	Details
NIF 2.0 Validator Tool@en	Details
NIF 2.0 Stanford Core Implementation@en	Details

Demo	
NIF Conversion Demo Service@en	Details
NIF Combinator Demo@en	Details

NIF Combinator: Combining NLP Tool Output

This web site show cases and combines available demo nif web services. Note that the list displayed here is not complete. More tools will be added soon. Read the [full documentation](#) [Bibtex]. The code for this demo is available [here](#). This service also allows you to retrieve the raw data. Use the *format* parameter which supports *rdxml*, *turtle*, *rdffson*, *ntriples* as values. You can also use the *Accept*: header with the appropriate content-type. Note that this service is currently using NIF-2.0 specification.

Action

- ☒ display and merge
- ☐ validate
- ☐ summarize

URI Recipe

- ☒ offset
- ☐ context-hash

Prefix

http://prefix.given.by/theClient#

Tools

- ☒ [Snowball Stemmer](#)
- ☐ [Stanford CoreNLP](#)
- ☐ [OpenNLP](#)
- ☒ [DBpedia Spotlight](#) (Confidence:)

Other NIF service:

Coming soon:

- ☐ [UIMA](#)
- ☐ [Mallet](#)
- ☐ [Gate ANNIE](#)
- ☐ [DBpedia Spotlight](#)

 reset

 submit

Input text

President Obama on Monday will call for a new minimum tax rate for individuals making more than \$1 million a year.

Merged Output

```
|@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rlog: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/rlog#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .

<http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#RFC5147String> rdf:type owl:Class .

<urn:uuid:fd2b4fd5-3974-4835-8469-73e95a93fe74> rdf:type rlog:Entry ;
    rlog:date "2014-10-15T09:55:43.207Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> ;
    rlog:level <http://persistence.uni-
```

Example 1: Snowball Stemmer Wrapper

- Stemming algorithm is a process for removing suffixes from words.
 - CONNECT
 - CONNECTED
 - CONNECTION
 - CONNECTING
 - CONNECTIONS

- 1. Open the USB stick folder
- 2. Go to “NIF_tutorial_hands_on_jars” folder
- 3. Open the “instructions.txt” file in a text editor
- 4. Open a terminal
- 5. Go to the “jar” folder

- Copy the second command of the instructions.txt
 - “java -jar snowball.jar -f text -i 'My favorite actress is Natalie Portman.'”
 - -f is used to define the format
 - -i is used to define the input
- Paste in the terminal

```
ciro@ciro:~/NIF_tutorial_hands_on_jars/jar$ java -jar snowball.jar -f text -i 'I am connected.'
```

```
ciro@ciro:~/NIF_tutorial_hands_on_jars/jar$ java -jar snowball.jar -f text -i 'I am connected.'
```

```
<http://cli.nlp2rdf.org/snowball#char=5,14>
  a                nif:Word , nif:RFC5147String ;
  nif:anchorOf      "connected" ;
  nif:beginIndex    "5" ;
  nif:endIndex      "14" ;
  nif:nextWord       <http://cli.nlp2rdf.org/snowball#char=14,15> ;
  nif:previousWord   <http://cli.nlp2rdf.org/snowball#char=2,4> ;
  nif:referenceContext <http://cli.nlp2rdf.org/snowball#char=0,15> ;
  nif:sentence       <http://cli.nlp2rdf.org/snowball#char=0,15> ;
  nif:stem           "connect" .
```

```
ciro@ciro:~/NIF_tutorial_hands_on_jars/jar$ java -jar snowball.jar -f text -i 'I am connected.'
```

NIF Offset

NIF Standard Annotations

```
<http://cli.nlp2rdf.org/snowball#char=5,14>
a nif:Word , nif:RFC5147String ;
nif:anchorOf "connected" ;
nif:beginIndex "5" ;
nif:endIndex "14" ;
nif:nextWord <http://cli.nlp2rdf.org/snowball#char=14,15> ;
nif:previousWord <http://cli.nlp2rdf.org/snowball#char=2,4> ;
nif:referenceContext <http://cli.nlp2rdf.org/snowball#char=0,15> ;
nif:sentence <http://cli.nlp2rdf.org/snowball#char=0,15> ;
nif:stem "connect" .
```

```
ciro@ciro:~/NIF_tutorial_hands_on_jars/jar$ java -jar snowball.jar -f text -i 'I am connected.'
```

NIF Offset

NIF Standard Annotations

Snowball Stem

```
<http://cli.nlp2rdf.org/snowball#char=5,14>
a nif:Word , nif:RFC5147String ;
nif:anchorOf "connected" ;
nif:beginIndex "5" ;
nif:endIndex "14" ;
nif:nextWord <http://cli.nlp2rdf.org/snowball#char=14,15> ;
nif:previousWord <http://cli.nlp2rdf.org/snowball#char=2,4> ;
nif:referenceContext <http://cli.nlp2rdf.org/snowball#char=0,15> ;
nif:sentence <http://cli.nlp2rdf.org/snowball#char=0,15> ;
nif:stem "connect" .
```

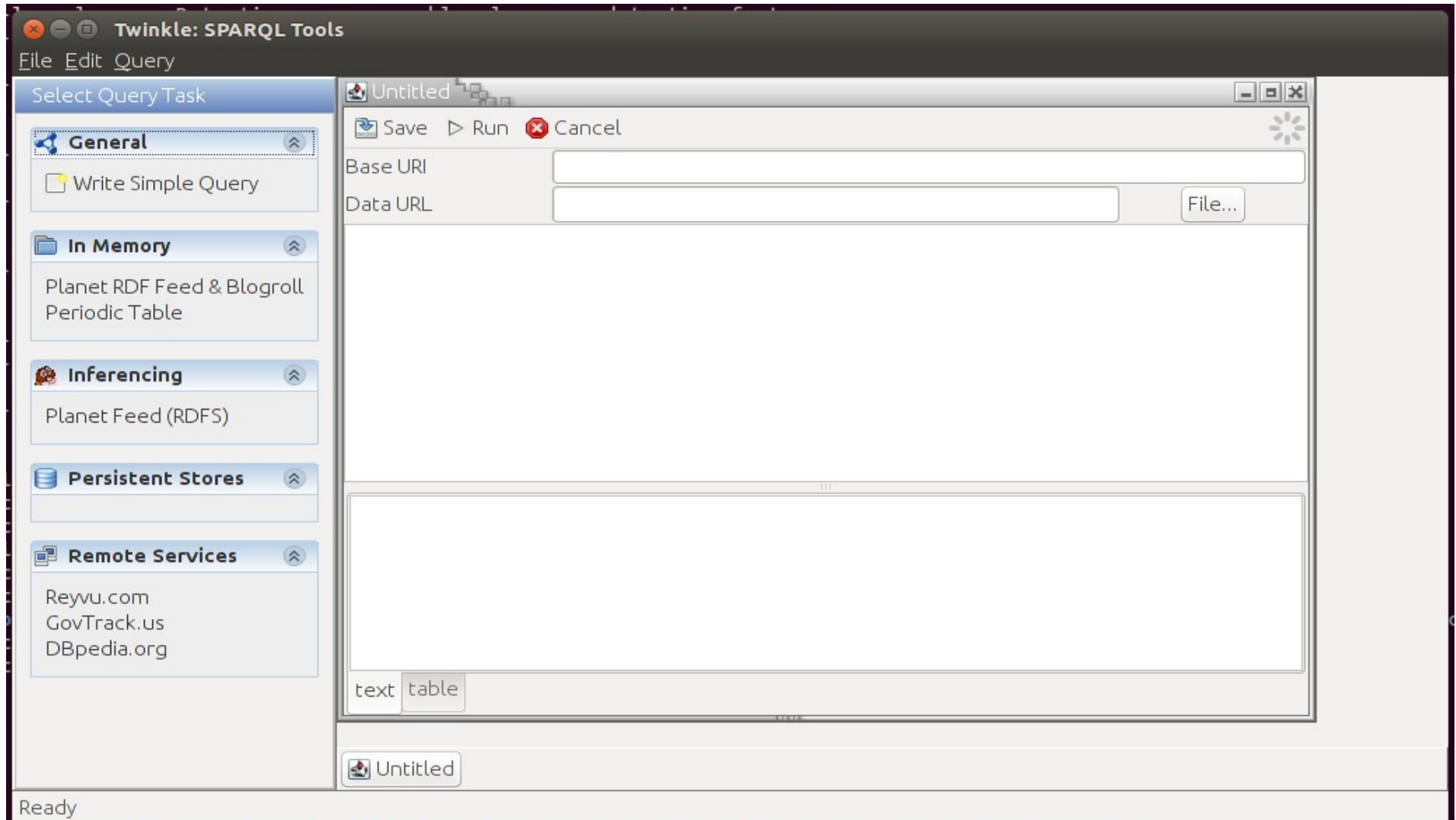

- Back to the terminal and use the first command of the instructions.txt

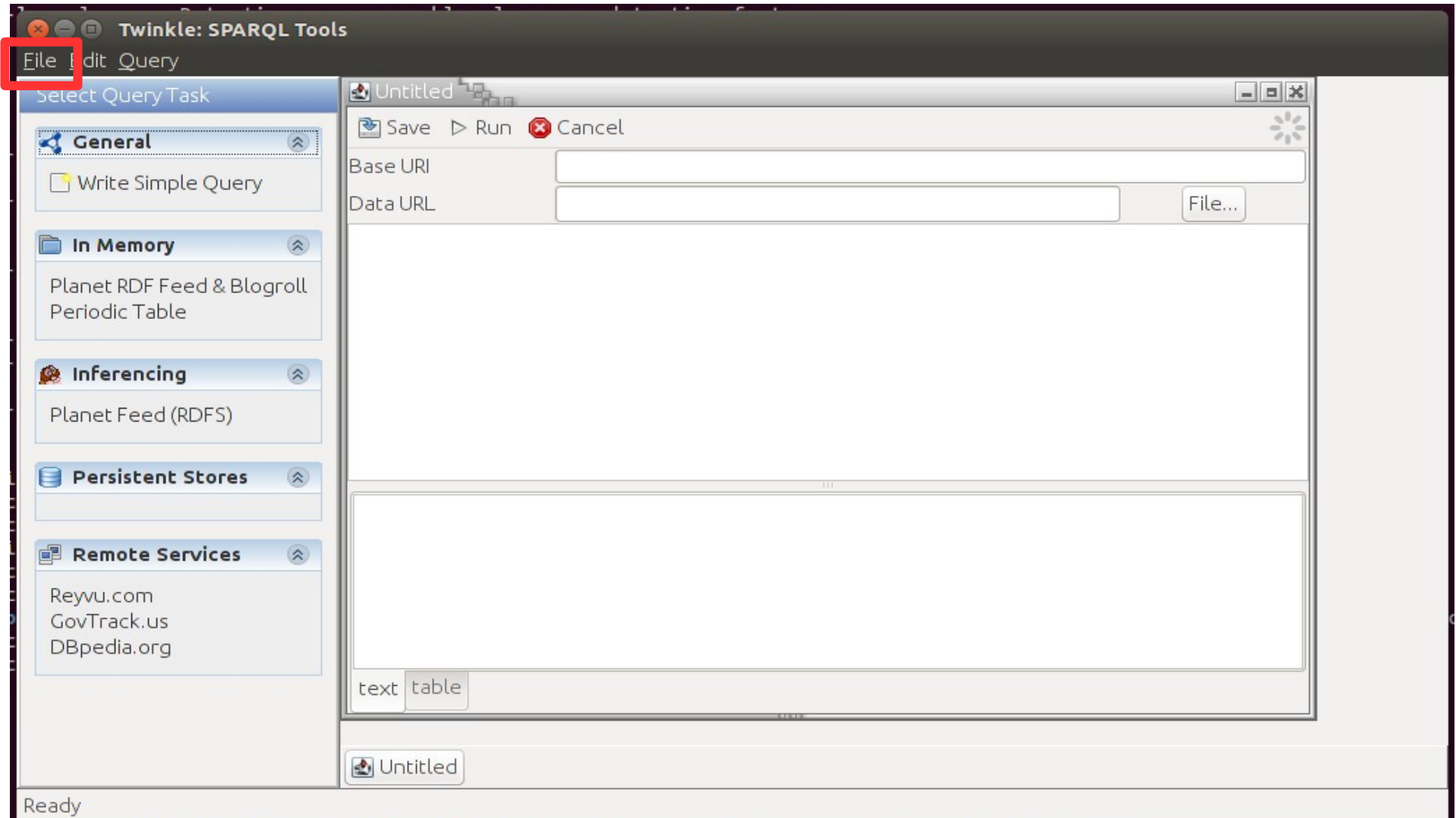
```
java -jar opennlp.jar -f text -i 'My favorite actress is Natalie Portman.' -modelFolder ../model/
```
- The -modelFolder parameter set the folder that contains the POS tagging OpenNLP trained models and tokenization.
- You might add the parameter “--outfile myAnnotatedFile.ttl” to store the triples in a file.

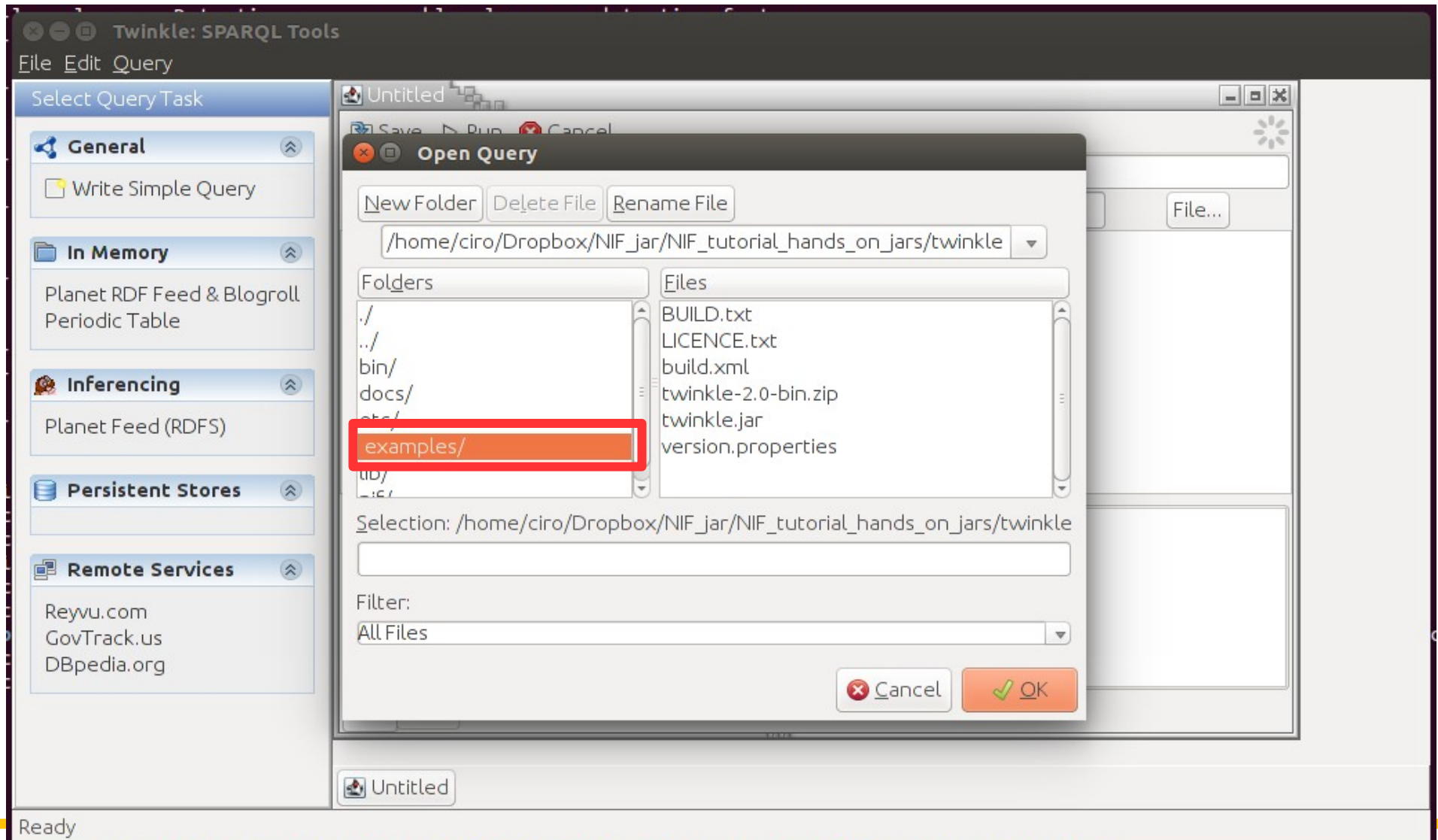
Example 2: Query Brown Corpus

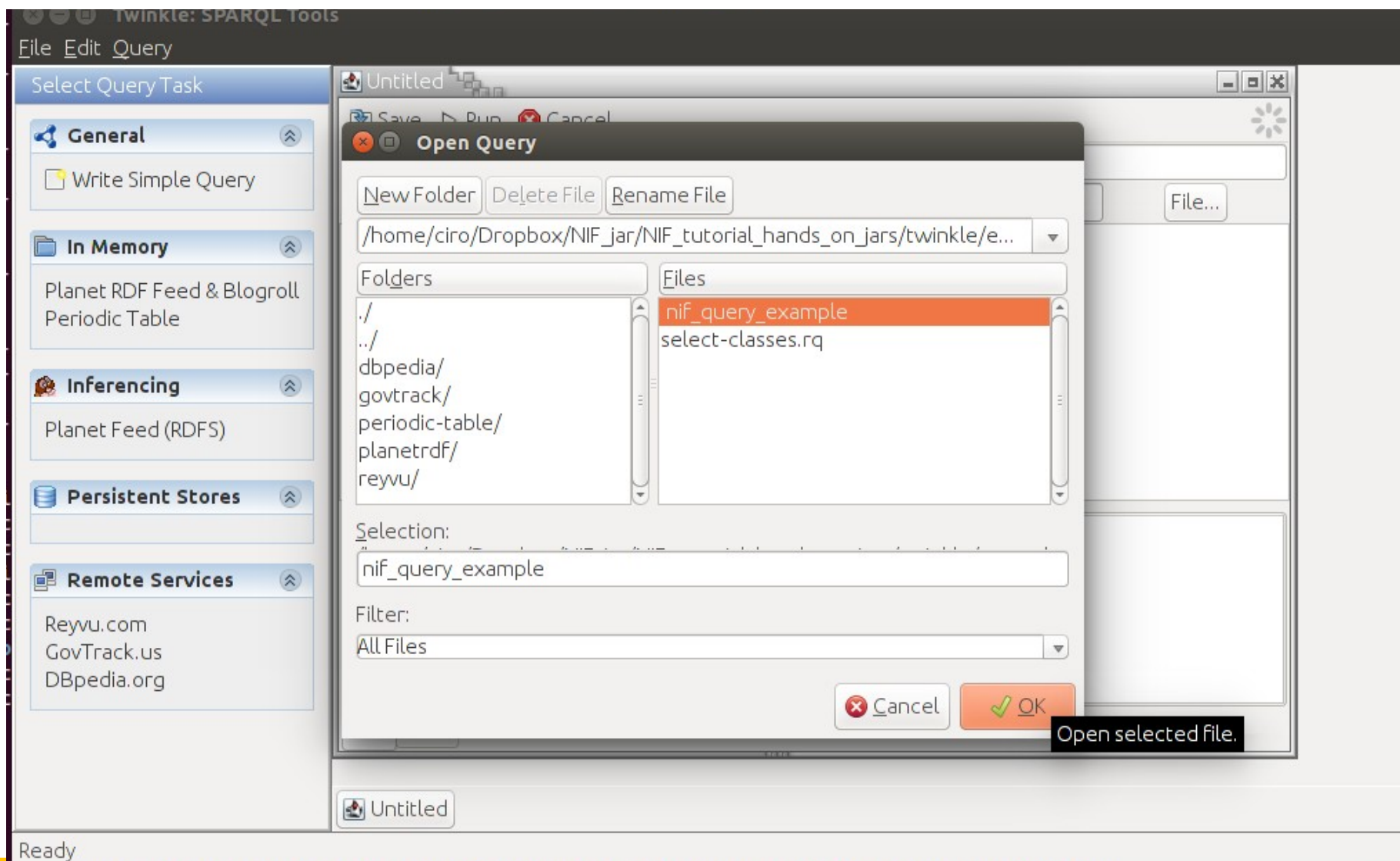
- Open the “/twinkle/example” folder
- Open the NIF_query_example file in a text editor and copy the query
- Open the “/twinkle” folder and run the command:

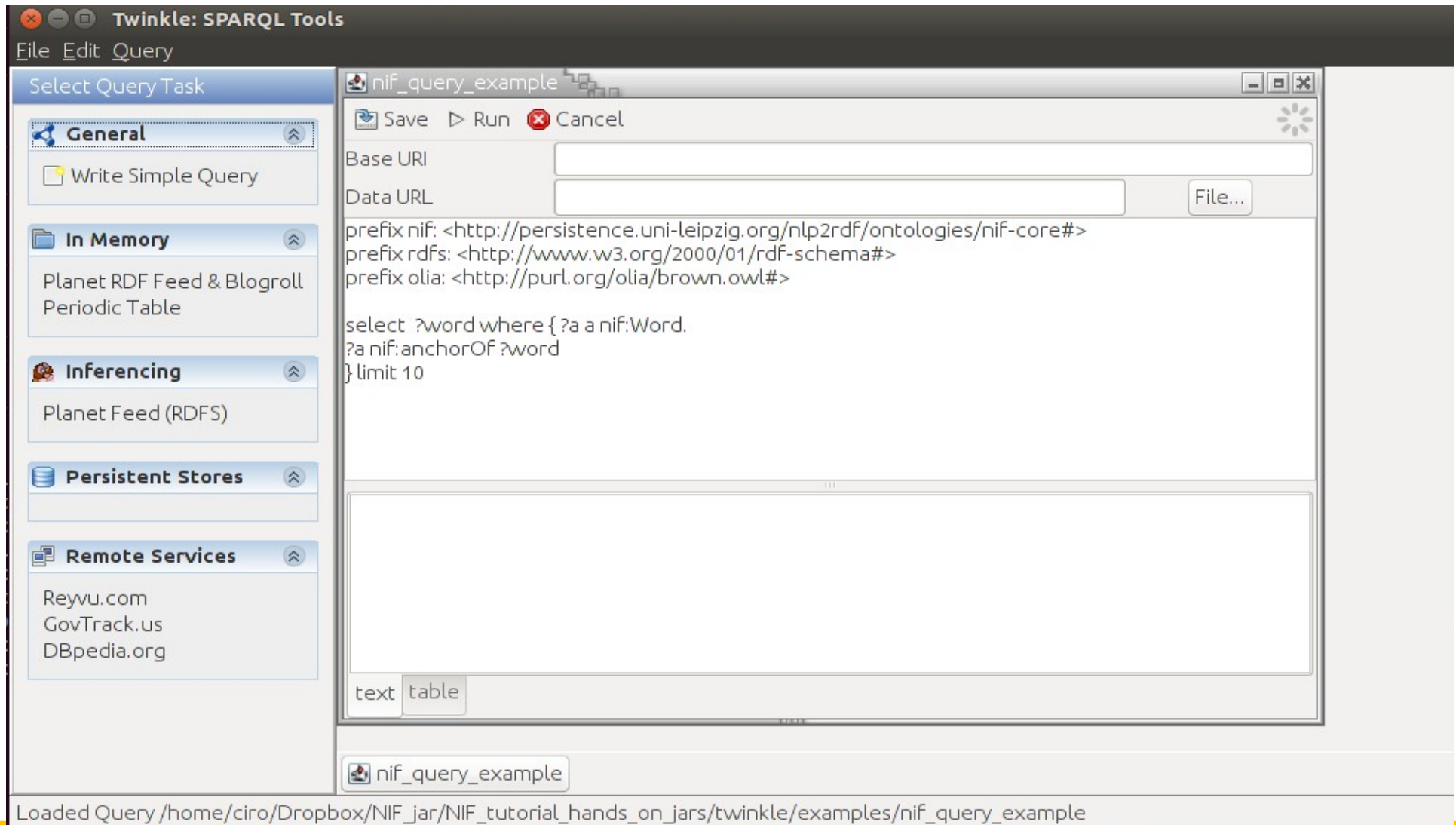
```
java -jar twinkle.jar
```

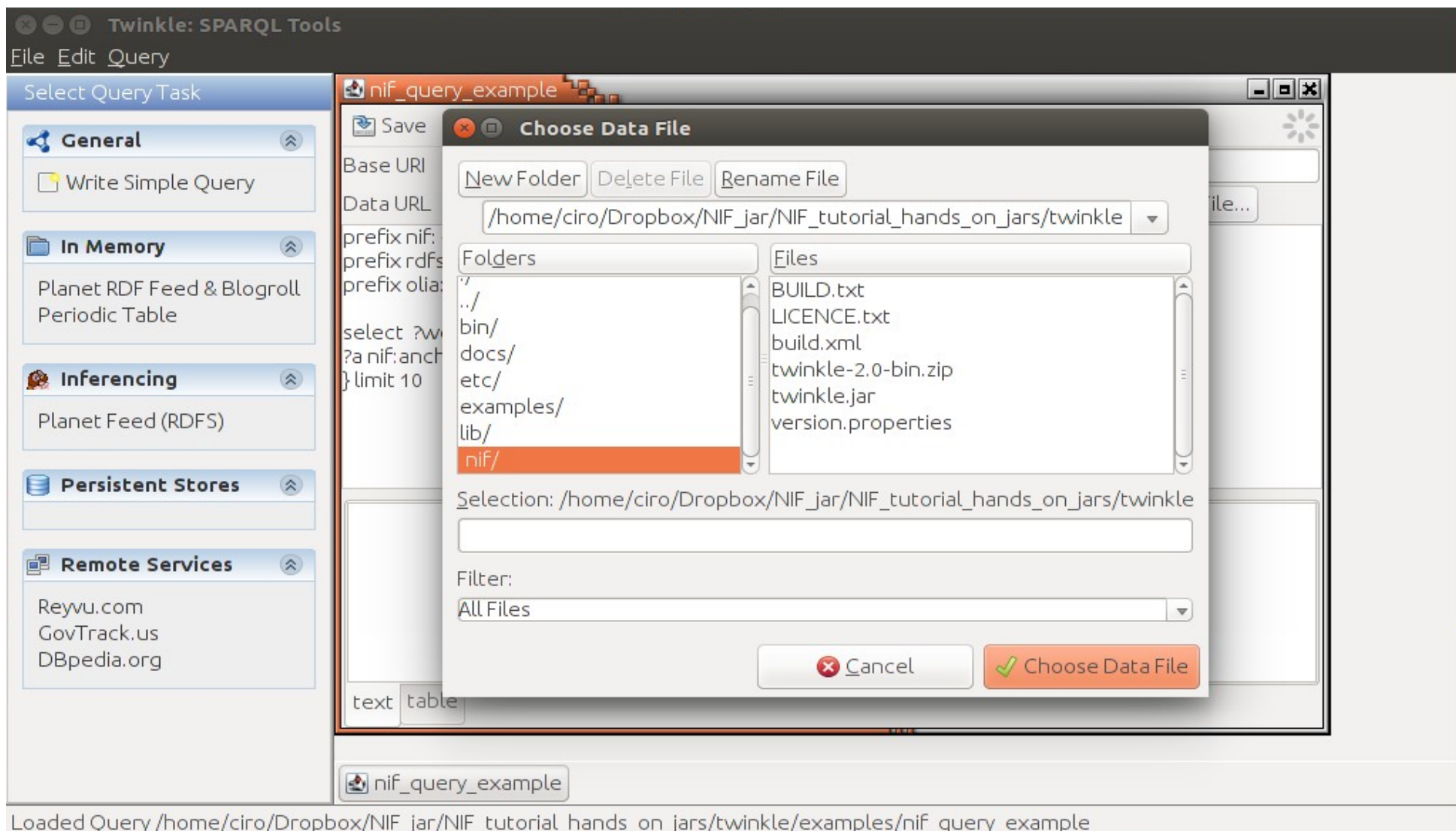


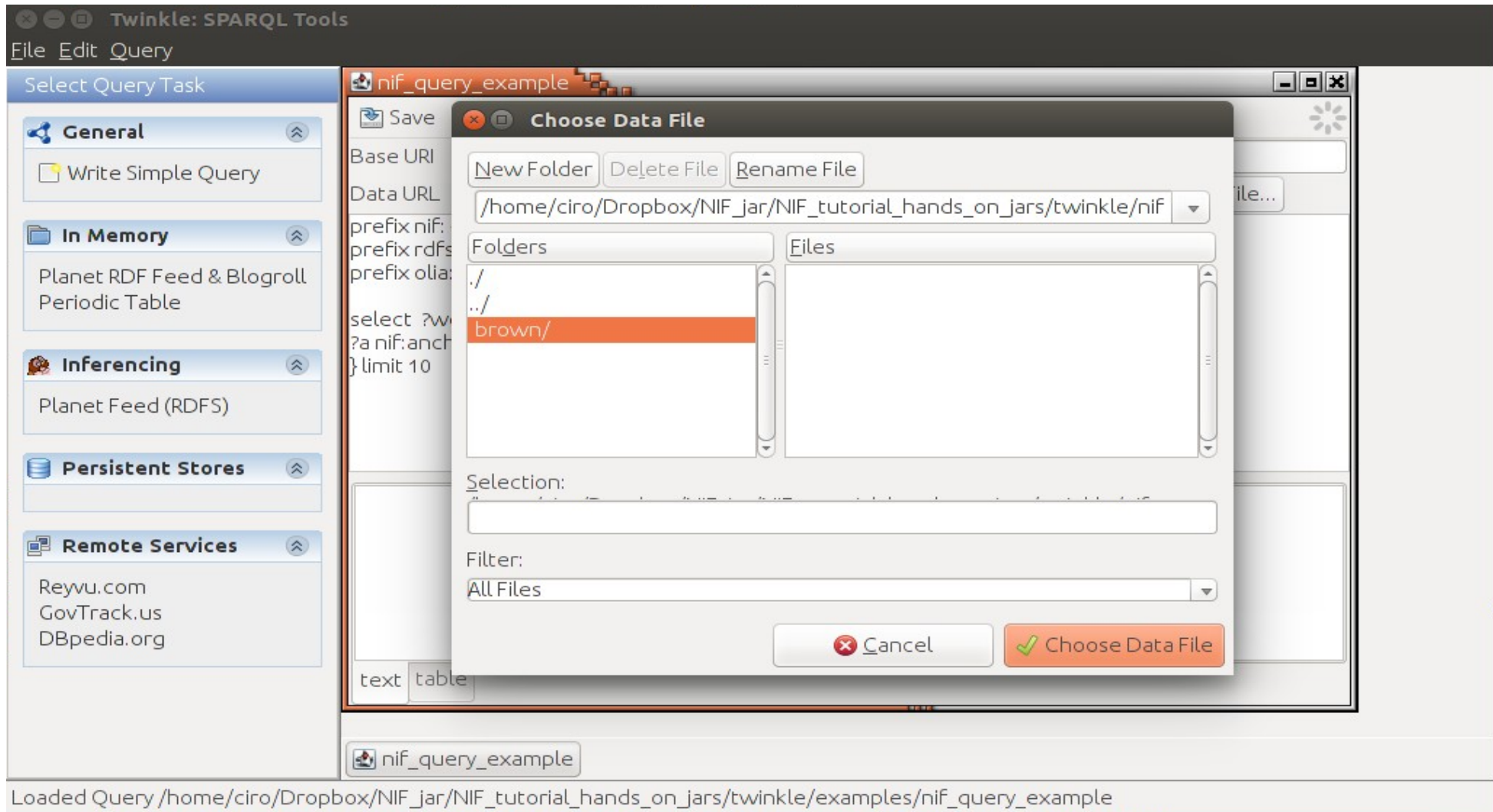


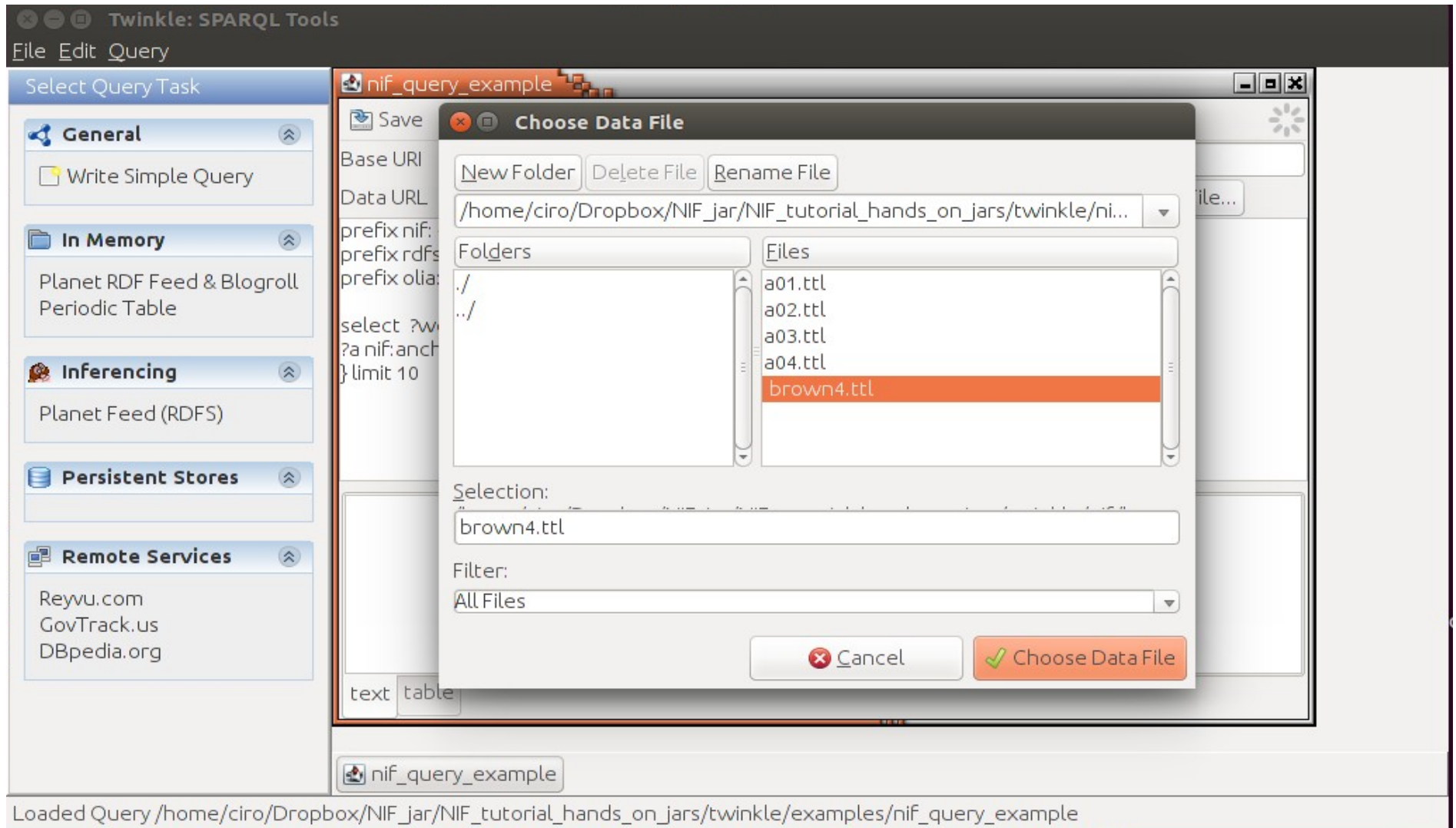


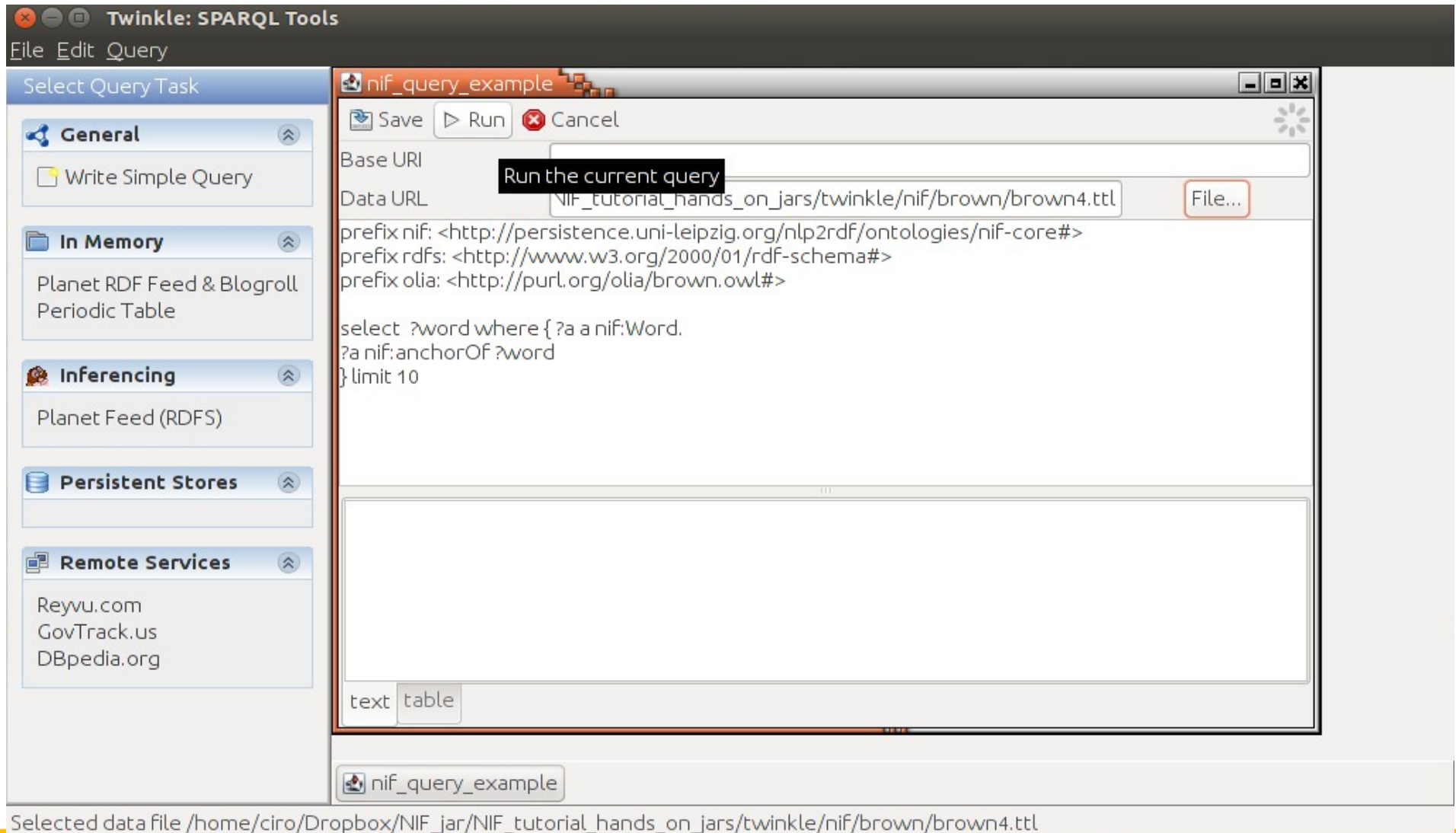












The screenshot shows the Twinkle: SPARQL Tools application. On the left is a sidebar with categories: General, In Memory, Inferencing, Persistent Stores, and Remote Services. The main window is titled 'nif_query_example' and contains a query editor. A black box with the text 'Run the current query' points to the 'Run' button. The query editor shows the following content:

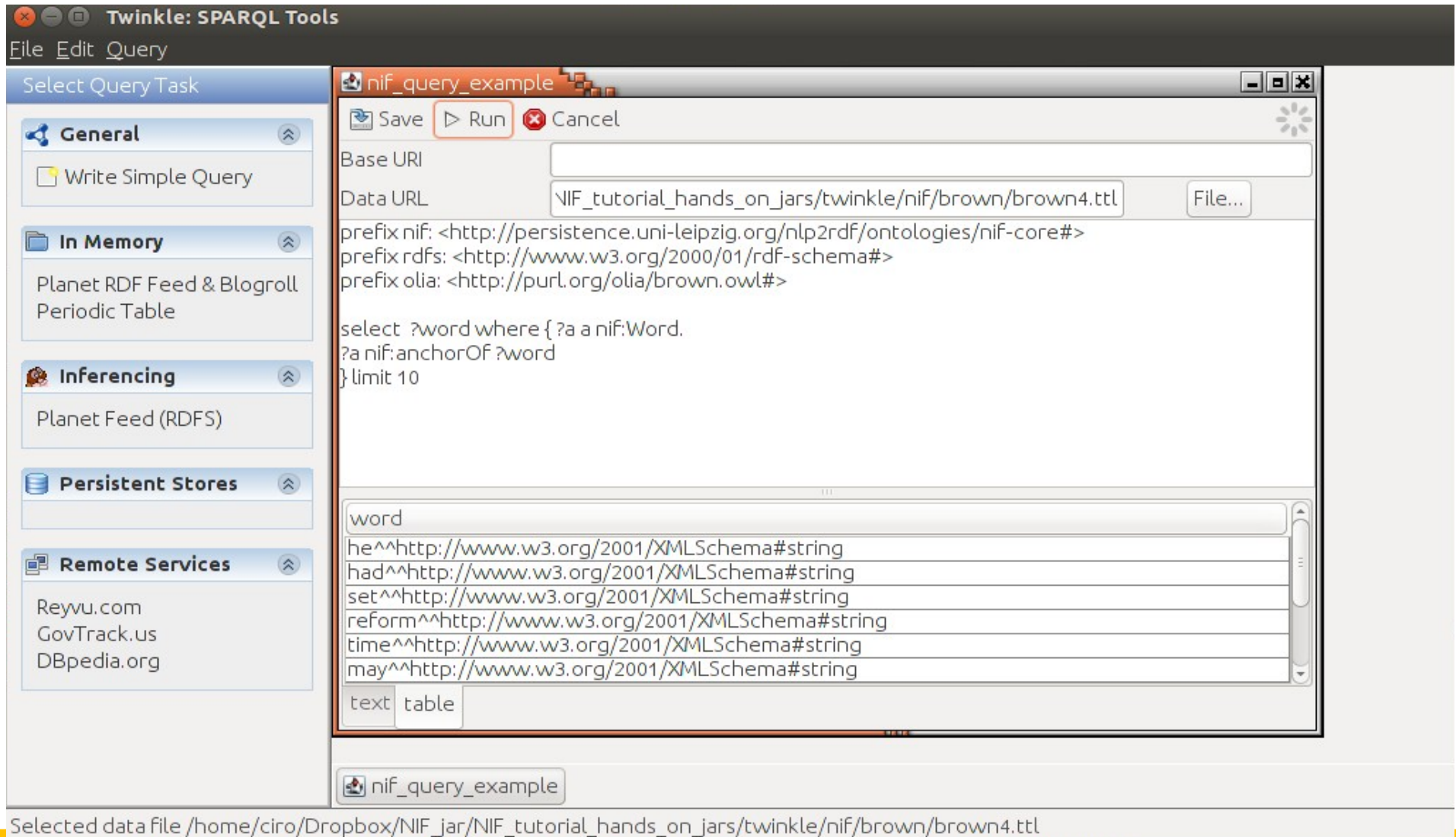
Base URI:

Data URL:

prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
 prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 prefix olia: <http://purl.org/olia/brown.owl#>

select ?word where {
 ?a a nif:Word.
 ?a nif:anchorOf ?word
} limit 10

Below the query editor is a large empty box for results, with 'text' and 'table' tabs at the bottom. At the very bottom of the application, a status bar reads: 'Selected data file /home/ciro/Dropbox/NIF_jar/NIF_tutorial_hands_on_jars/twinkle/nif/brown/brown4.ttl'.



The screenshot shows the Twinkle: SPARQL Tools application. On the left is a sidebar with categories: General (Write Simple Query), In Memory (Planet RDF Feed & Blogroll, Periodic Table), Inferencing (Planet Feed (RDFS)), Persistent Stores, and Remote Services (Reyvu.com, GovTrack.us, DBpedia.org). The main window is titled 'nif_query_example' and contains a 'Run' button. Below the button are fields for 'Base URI' and 'Data URL' (set to 'NIF_tutorial_hands_on_jars/twinkle/nif/brown/brown4.ttl'). A 'File...' button is next to the Data URL. The query text is:

```
prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix olia: <http://purl.org/olia/brown.owl#>

select ?word where {
  ?a nif:Word.
  ?a nif:anchorOf ?word
} limit 10
```

Below the query is a table of results:

word
he^^http://www.w3.org/2001/XMLSchema#string
had^^http://www.w3.org/2001/XMLSchema#string
set^^http://www.w3.org/2001/XMLSchema#string
reform^^http://www.w3.org/2001/XMLSchema#string
time^^http://www.w3.org/2001/XMLSchema#string
may^^http://www.w3.org/2001/XMLSchema#string

At the bottom, a status bar indicates the selected data file: /home/ciro/Dropbox/NIF_jar/NIF_tutorial_hands_on_jars/twinkle/nif/brown/brown4.ttl.

Exercise 3: Querying your own NIF annotated string

Querying your own NIF annotated string

1. Annotate your string using one of the wrappers
2. Save your annotated sentence to a file (using “--outfile”)
3. Open Twinkle
4. Query your string using Twinkle

- Query your annotated string:
 - nif:Context
 - nif:Sentence
 - nif:anchorOf
 - nif:oliaCategory
 - nif:oliaLink
- ... or practice with Brown Corpus!

Thank you!

<http://site.nlp2rdf.org/>

NLP2RDF Google+ Community 