A Lightweight String Based Method of Encoding Etymologies in Linked Data Lexical Resources

Anas Fahad Khan
CNR-ILC
Italy
fahad.khan@ilc.cnr.it

Maxim Ionov
University of Zaragoza
Spain
mionov@unizar.es

Paola Marongiu
CNR-ILC
Italy
paola.marongiu@ilc.cnr.it

Ana Salgado
NOVA CLUNL
Portugal
anasalgado@fcsh.unl.pt

Abstract

In this submission, we propose an approach to encoding etymological information as strings with formal syntax ("etymology strings"). We begin by discussing the advantages of such an approach compared to modelling etymologies and etymons explicitly as RDF individuals. Next we give a formal description of the regular language underlying our approach as an Extended Backus-Naur Form grammar (EBNF). We use the Chamuça Hindi lexicon as a test case for our approach and show a practical application of the approach using SPARQL queries that extract necessary information from etymological strings.

1 Introduction

Linked Data best practices encourage the use of HTTP URI's to name things and representing every bit of information as RDF triples (statements) based on a formally defined data model. In other words, the preference is for modelling data explicitly in the form of knowledge graphs in which all or most of the entities are represented as RDF classes or individuals each with its own individual URI (individuals can be represented as blank nodes, although this limits their usability). However, certain kinds of data do not lend themselves easily to being modelled this way. This is the case for descriptions of dynamic or temporal phenomena: these latter are most naturally modelled via the addition of a temporal parameter to standard RDF subject-predicate-object triples something which, in general, can only be done via workarounds, e.g., via the reification of properties. Linguistic Linked Data offers many examples of such dynamic phenomena, notably in the form of etymologies, i.e, hypothetical word histories (Khan, 2020). Indeed, when it comes to etymologies, aside from the requirement to represent temporality we have a number of other modelling considerations that make the comprehensive description of such resources potentially very complicated from an RDF point of view. For instance, the following:

- The requirement to represent the hypothetical status of etymologies. In many cases etymologies carry with them a high level of uncertainty and many works will often provide more than one etymology for the same word.
- Related to the previous requirement, in a large number of cases it is important to be able to represent references to the scholarly literature/dictionaries, corpus citations, etc.

In addition, in order to carry out a 'deep' modelling of etymologies, we should, according to linked data best practices, create URIs/individuals for etymons and cognates as well as for etymologies themselves and for cognate sets, that is, to reify all of these elements, as well as adding other elements in order to represent the temporal duration of relationships and properties between individuals. A vocabulary that meets all or most of these requirements will end up being complicated and difficult to use, going well beyond the basic constructions and elements of languages such as RDFS and OWL or more specialsed vocabularies such as OntoLex (see for instance the proposal given in (Khan, 2018)). In addition, such a vocabulary would be hard to create in a theory-agnostic way hence it might end up being unusable for some resources. At the same time, these are fairly standard requirements to cover phenomena that are found in etymological descriptions provided by a lot of lexicographic resources. However, in a large number of use-cases not all the complexity is warranted: What we are looking for are shallow descriptions of etymologies which are of limited complexity and that lend themselves to fast querying and/or processing.[1]

[1]Note that full modelling can still be integrated with this approach using SPARQL UPDATE queries given that there is a vocabulary that can accommodate that.

Consequently, in the current work, instead of proposing a standard, 'deep' RDF based modelling, we propose the use of strings to model etymologies in a 'shallow' way. In order to help ensure interoperability and to facilitate querying of such strings in SPARQL, we define a regular language for representing such 'etymological strings', one that is based on textual conventions for the representations of etymologies in lexicographic works as well as in other kinds of literature.

The rest of the paper is structured as follows. In Section 2 we motivate the use of our approach and provide further details of the kinds of use cases to which we recommend applying this approach. Next, in Section 3 we give a full description of the regular language which we propose as a solution. Afterwards, in Section 4 we give the description of a use case with which we illustrate the way we recommend to query data modelled using this kind of language and present a web interface that uses this approach.

2 Motivation and Use Cases

This work aims to provide a lightweight method of encoding etymological information as string literals. This is particularly useful when (i) the original information is fairly simple; (ii) only a shallow representation of the etymology is required; (iii) a more involved kind of RDF modeling would introduce unnecessary complexity or overhead (and we may not want to e.g., explicitly represent etymologies as hypotheses or model time as a parameter). For instance, in many cases source etymologies will be given in a form similar to the following one for the word *friar* (example from (Durkin, 2009)):

> Latin *frāter* "brother" > Old French *frère* "brother, member of a religious order" > Middle English *frere*, *friar* > Modern English *friar*

where '>' is a standard symbol that marks the direction of the etymological development of the target word (in this case Modern English *friar*).

This example gives a description of the history of a word, but etymologies can also describe other linguistic elements, such as senses as in the following case where the semantic development of the English word is given *sad* (example taken, again, from (Durkin, 2009)):

> Satisfied, having had one's fill (of something) [metamorphized and narrowed] > weary or tired (of something) [borrowed] > sorrowful, mournful

Our argument is that in these, and a large number of similar cases, following a more involved RDF modelling (such as that proposed in (Khan, 2018)) would create too much overhead in terms of RDF triples, given that most often users are interested only in basic kinds of etymological information or an entry as a whole. Indeed, in such simple cases, we could encode the essential information in the form of a string literal allowing users to extract necessary pieces of information via SPARQL queries that make use of regular expressions or via the use of simple string matching functions on the results of a SPARQL query, thus providing complexity-on-demand while preserving all the information. An alternative to this approach (that would also avoid the prolixity which we have just mentioned) would be to define an RDF vocabulary, or series of vocabularies, that captured only some of the requirements which we listed above but that e.g., didn't take the explicit representation of time into consideration or that associated only limited kinds of information with different individual stages of an etymology. However, we feel ours is a cleaner alternative, and one that better fits the current trend towards minimal computing (and of course in more complicated kinds of use cases we would suggest using a more extensive ontology based approach).

In order to preserve interoperability as far as possible, our idea is to define a simple regular language (i.e., a set of strings that can be described with a regular expression) which our etymological strings must belong to. This regular language is based on the simple string representation used in etymological dictionaries, and example of which we saw above. We describe this language in the following section in EBNF for the purposes of explanation.

3 Description of the Etymological String Regular Language

In the rest of the article we lay out and describe our first proposal for an etymological string language. This language has been designed with the following features:

- It allows one or more complete etymologies in a single string, each of which is separated by the ‘|’ symbol; these are alternative etymologies for the lexical element in question (nesting is not possible in a regular language, potentially there could be a context-free superset of this language that allows a shorthand for this using brackets). Each of these etymologies can be associated with a bibliographic source between parentheses.

- Each step is an etymology separated by a ‘>’ symbol and individual steps adhere to the format of: *language code* followed by one or more alternative forms for a *lemma* followed by one or more *senses* separated by an ampersand ‘&’.
- We allow for an additional specification of each step with a transition note between square brackets.

According to this language, our previous *friar* example can be rewritten as follows:

> lat frater > fro frere ‘brother’ & ‘also member of a religious order of ‘brothers’’ > enm friar, frere > eng friar (Source: Durkin) .

Similarly we can rewrite the *sad* example above as follows:

> ‘Satisfied, having had one’s fill (of something)’ [metamorphized and narrowed] > ‘weary or tired (of something)’ [borrowed] > ‘sorrowful, mournful’

This language is clearly limited in the kinds of etymological information which can be captured. At the same time it fits a large number of simpler cases as found in many lexicographic works. The clear benefit of the formalism is that it provides a middle ground between human- and machine-readability: while still looking familiar to lexicographers and etymologists, it can be validated and processed with efficient and simple computational methods.

EBNF Grammar

In this section we present an EBNF grammar of our regular language[2]. This permits us to give a precise formal definition of our language in a fairly (for humans) understandable and transparent way, something that would not be the case if we presented our language as one long regular expression.

```
etymologies = etymology , { "| " , etymology } , [ " " ],"."
      ;

etymology = step , { [ transition_note ] , ">" , [ " " ] ,
      step } ;

transition_note = "[" , printable_no_quote_seq , "]" ;

step = [ lang , " " ] , ["? ", " "],
      [ lemmas , " " ] ,
      [ senses , " " ] ,
      [ source ] ;

lang = letter , { letter | "-" } ;

lemmas = lemma , { ", " , lemma } ;

lemma = letter , { letter } ;

senses = "‘" , sense , "’" , { " & " , "‘" , sense , "’" } ;

sense = printable_no_quote_seq ;

source = "(Source:" , printable_no_quote_seq , ")" ;

letter = ? UnicodeLetter ? ;

printable_no_quote_seq = printable_no_quote , {
      printable_no_quote } ;

printable_no_quote = ? isPrintable & notQuote ? ;
```

As will be seen, our language already allows us to capture a lot of different kinds of etymologies (as hopefully the next section will demonstrate). However there may be small elements which will be added in subsequent versions to make the formalism even more useful.

4 The CHAMUÇA Test Set

As a test set for our regular language we decided to use the CHAMUÇA Hindi language lexicon (CHAMUÇA-Hi), one of the outputs of an ongoing project, CHAMUÇA, which aims to trace the impact of Portuguese on the languages of Asia (Khan et al., 2024). CHAMUÇA-Hi consists of just over a hundred entries in Hindi, each of which at least plausibly derive from an original Portuguese etymon.[3] For many of these entries there are alternative etymologies positing a non-Portuguese origin of the entry in question. Overall, however, the etymological information included in the dataset is fairly shallow; it is therefore ideal for showcasing our lightweight approach.

We had already converted our dataset into RDF

[2]Note that although we have used EBNF to present our langauge it is regular without introducing features that would place it higher in the Chomsky hierarchy.

[3]The lexicon can be downloaded here: https://github.com/anasfkhan81/Etymologies_as_Strings/blob/main/chamuca_hi_lex.ttl.

(Khan et al., 2024), but decided to generate another version with the addition of etymological strings which follow the regular language proposed in this article, associating the entries with their etymologies using the lexinfo[4] `etymology` property. For instance, see the following entry for the word अन-न्नास (anannaas) meaning ‘pineapple’:

```
अनन्नास:_entry a ontolex:LexicalEntry,
   ontolex:Word ;
lexinfo:domain <http://lari-datasets.ilc.cnr.it/
  chadoms#botany> ;
lexinfo:etymologicalRoot <http://lari-datasets.ilc.cnr.
  it/chamuca_pt_lex#ananás> ;
lexinfo:etymology "tpn naná ''pineapple (Source:
  Wiktionary) > pt ananás ''pineapple (Source:
  Dalgado) ." ;
lexinfo:gender lexinfo:masculine ;
lexinfo:partOfSpeech lexinfo:commonNoun ;
frac:frequency [ a frac:Frequency ;
      rdf:value 0 ;
      frac:observedIn :hiTenTen21 ] ;
ontolex:canonicalForm अनन्नास:_lemma ;
ontolex:lexicalForm अनन्नास:अनन्नास_dp_form_,
   अनन्नास:अनन्नास_os_form_,
   अनन्नास:अनन्नास_vs_form_,
   अनन्नासो:अनन्नास_vp_form_,
   अनन्नासों:अनन्नास_op_form_ ;
ontolex:sense अनन्नास:_sense .
```

Other examples of etymological strings from the dataset include:

```
"pt ? carabina (Source:
Dalgado) | French carabine
(Source: Dalgado) | fa qarābīn
(Source: McGregor) ."
```

and

```
"pt ? baptismo (Source:
Dalgado) | en baptism (Source:
Dalgado) ."
```

The question mark at the beginning of each lemma here signals the fact that the etymology is regarded as being doubtful in the source itself.

In each of these cases, the creation of an RDF graph explicitly encoding the same etymological information would have meant creating individuals for each of these etymons as well as for the etymology itself with a high cost in the number of resulting triples. Encoding the information as strings as we have done in this case, we are still able to extract quite a lot of relevant etymological information either via SPARQL queries or using basic string processing functions from different programming languages. For instance we can write a simple query which gives the number of alternative entries for each etymology:

```
PREFIX lexinfo: <http://www.lexinfo.net/ontology/2.0/
    lexinfo#>
PREFIX ontolex: <http://www.w3.org/ns/lemon/ontolex
    #>

SELECT ?entry
    ((STRLEN(STR(?etymology)) - STRLEN(REPLACE(
    STR(?etymology), "\\|", ""))) + 1 AS ?
    numAlternatives)
WHERE {
  ?entry a ontolex:LexicalEntry ;
      lexinfo:etymology ?etymology .
}
```

Another example would be a query which searches of all words which potentially have an ancient Greek etymon:

```
PREFIX lexinfo: <http://www.lexinfo.net/ontology/2.0/
    lexinfo#>
PREFIX ontolex: <http://www.w3.org/ns/lemon/ontolex
    #>

SELECT ?entry ?etymology
WHERE {
  ?entry a ontolex:LexicalEntry ;
      lexinfo:etymology ?etymology .

  # grc in the beginning or after >
  # or after |
  FILTER (
    REGEX(STR(?etymology),
        "(^|>|\\|)\\s*grc\\b")
  )
}
```

Using a combination of queries like this users can extract specific parts of etymologies in which they are interested. An additional advantage of this approach is that the users are not limited to using SPARQL queries to process the data: while etymological strings has to be extracted with SPARQL, further processing can be done in a variety of ways, thanks to wide support of regular expressions by software and programming languages.

To demonstrate a possible way to use etymological strings in a user-friendly way, we created a web interface that provides basic etymological data for a chosen entry from this dataset.[5]

5 Conclusions and Future Work

In this article we have presented a first draft of our work on etymological strings, a formalised way to represent etymological information in a string,

[4] https://lexinfo.net/

[5] https://anasfkhan81.github.io/Etymologies_as_Strings/.

without expanding it to complex graph representations. While unconventional, it is a convenient and efficient way to hide unnecessary data complexity without losing it, all while providing the data in both machine- and human-readable way.

In the future we plan to enrich these strings with further kinds of information such as e.g., part of speech and gender. However we feel that the proposed formal language is already suitable for a large number of use cases. However we do not want to make the language or strings too complicated since this would defeat the purpose of using our approach in the first place. Also in the future we plan to look into whether our approach is also useful for other kinds of information, i.e., morphological information.

Acknowledgments

The work is in part supported by the H2IOSC Project - Humanities and cultural Heritage Italian Open Science Cloud funded by the European Union NextGenerationEU - National Recovery and Resilience Plan (NRRP) - Mission 4 “Education and Research” Component 2 “From research to business” Investment 3.1 “Fund for the realization of an integrated system of research and innovation infrastructures” Action 3.1.1 “Creation of new research infrastructures strengthening of existing ones and their networking for Scientific Excellence under Horizon Europe” - Project code IR0000029 - CUP B63C22000730005. Implementing Entity CNR.

References

Philip Durkin. 2009. *The Oxford Guide to Etymology*. OUP Oxford.

Anas Fahad Khan. 2018. Towards the Representation of Etymological Data on the Semantic Web. *Information*, 9(12):304.

Anas Fahad Khan. 2020. Representing Temporal Information in Lexical Linked Data Resources. In *Proceedings of the 7th Workshop on Linked Data in Linguistics (LDL-2020)*, pages 15–22, Marseille, France. European Language Resources Association (ELRA).

Anas Fahad Khan, Ana Salgado, Isuri Anuradha, Rute Costa, Chamila Liyanage, John Philip McCrae, Atul Kumar Ojha, Priya Rani, and Francesca Frontini. 2024. Chamuça: Towards a Linked Data Language Resource of Portuguese Borrowings in Asian Languages. In *Proceedings of the 9th Workshop on Linked Data in Linguistics@ LREC-COLING 2024*, pages 44–48.