1. NIF WRAPPER FOR NERD

1.1 Status of NIF 1.0

The *NLP Interchange Format* (NIF) is an RDF/OWL-based format that aims to achieve interoperability between *Natural Language Processing* (NLP) tools, language resources and annotations. The NIF specification has been released in an initial version 1.0 in November 2011[1] and describes how interoperability between NLP tools, which are exposed as NIF web services can be achieved. Extensive feedback was given on several mailing lists[3] and a community of interest was created to improve the specification[4]. As of now implementations for 8 different NLP tools (e.g. UIMA, Gate ANNIE and DBpedia Spotlight) exist and a public web demo is available[5].

The motivation behind NIF is to allow NLP tools to exchange annotations about documents in RDF. Hence, the main prerequisite is that parts of the documents (i.e. strings) are referenceable by URIs, so that they can be used as subjects in RDF statements. We call an algorithm to create such identifiers URI TEMPLATE: For a given text t (a sequence of characters) of length $|t|$ (number of characters), we are looking for a *URI template* to create a URI, that can serve as a *unique* identifier for a substring s of t (i.e. $|s| \leq |t|$). We generally assume that this substring consists of adjacent characters only and is therefore a unique character sequence within the document, if we account for parameters such as context and position.

The NIF 1.0 specification provides two URI templates, which can be used to represent strings as RDF resources. We will focus here mainly on the first template using offsets. In the top part of Figure 1, two triples are given that use the following URI as subject:

`http://www.w3.org/DesignIssues/LinkedData.html#offset_717_729_Semantic%20Web`

According to the specification, the URI should be interpreted as refering to the *string* of the document reachable at `http://www.w3.org/DesignIssues/LinkedData.html` from the character index 717 until the index 729 (counting the gaps between the characters).

NIF 1.0 mandates that the whole string of the document has to be included in the output as an `rdf:Literal` to serve as the reference point. Within the framework of RDF and the current usage of NIF 1.0 for the interchange of output between NLP tools the definition of the semantics is sufficient to produce a working system. However, problems arise if additional interoperability with Linked Data or fragment identifiers[6] and ad-hoc retrieval of content from the Web is demanded. The actual retrieval method (such as content negotiation) to retrieve and validate the content for `#offset_717_729_Semantic%20Web` is left underspecified as is the relation of NIF URIs to fragment identifiers for MIME types such as text/plain (see RFC 5147[7]).

[1]Release: [2], Specification 1.0: `http://nlp2rdf.org/nif-1.0`
[3]**TODO:**which ones, ontolog forum, nlp2rdf
[4]community portal `http://nlp2rdf.org/get-involved`
[5]`http://nlp2rdf.lod2.eu/demo.php`
[6]The following Wikipedia article provides an extensive collection of links to standards and usages regarding fragment ids: `http://en.wikipedia.org/w/index.php?title=Fragment_identifier&oldid=439234353`
[7]`http://tools.ietf.org/html/rfc5147`

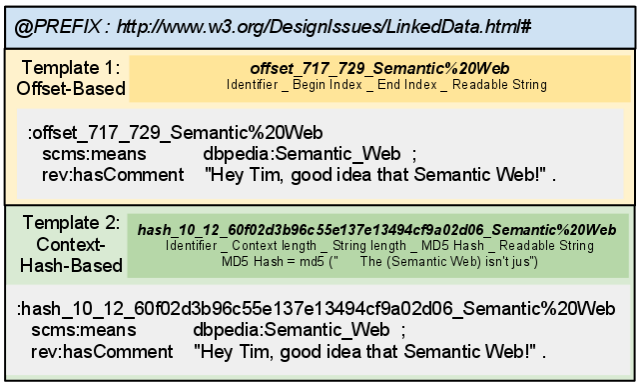

Figure 1: NIF URI templates: Offset (top) and context-hashes (bottom) are used to create identifiers for strings.

The triples below show how a string of a document is represented in NIF 1.0. A connection to the DBpedia entity "Semantic Web" is attached similar to "one entity per name" in NERD:

```
@prefix : <http://www.w3.org/DesignIssues/LinkedData
    .html#>
@prefix str: <http://nlp2rdf.lod2.eu/schema/string/>
    .
#the string of the document is represented by the
    URI and the actual string is included via the
    sourceString property.
:offset_0_26546_%3Chtml%20xmlns%3D%22http%3A%2F%2F
    rdf:type          str:Document ;
# connection to a substring
    str:subString     :offset_717_729_Semantic%20Web;
# [...] are all 26547 characters as rdf:Literal
    str:sourceString "[...]" .
# a substring of the document. The scms:means
    connection is comparable to the NERD - OEN
    definition.
:offset_717_729_Semantic%20Web a str:String;
    scms:means        dbpedia:Semantic_Web .
```

1.2 Changes to the core of NIF

Currently, the NIF specification is underspecified with regard to how exactly the string is retrieved from the web document and NIF 1.0 also does not formalise explicitly in which manner URIs refer to strings. According to the RDF semantics, if such an interpretation were formalised, it could be considered a "semantic extension"[8] of RDF, as "extra semantic conditions" are "imposed on the meanings of terms"[9]. A complete formalisation is still work in progress, but the idea is explained here: The NIF URIs will be grounded on Unicode Characters[10]. For all resources of type *str:String*, the universe of discourse will then be the powerset over the concatenation of Unicode characters. Perspectively, we hope that this will allow for an unambiguous interpretation of NIF by machines.

Furthermore, the class *str:Document* provided by NIF caused some confusion. The term "Document" is inappropriate as the real intention was to capture an arbitrary grouping of

[8]`http://www.w3.org/TR/2004/REC-rdf-mt-20040210/#urisandlit`
[9]`http://www.w3.org/TR/2004/REC-rdf-mt-20040210/#intro`
[10]especially Unicode Normalization Form C `http://www.unicode.org/reports/tr15/#Norm_Forms` counted in Code Units `http://unicode.org/faq/char_combmark.html#7`

characters forming a unit, which could also be applied to a *paragraph* or a *sentence* and is highly dependent upon the *context* in which the string is actually used. To appropriately capture the intention of such a class, we will distinguish between the notion of outside and the inside context of a piece of text. The inside context is easy to explain and formalise, as it is the text itself and therefore it provides a context for each substring contained in the text. The outside context is more vague and is given by an outside observer, who might arbitrarily interpret the text as a "book chapter" or a "book section" or whatever pleases him.

A new class str:Context now provides a clear reference point for all other relative URIs used in this context and blocks the addition of information from a larger (outside) context by definition. For example the new class str:Context is disjoint with foaf:Document as labeling a context object as a document is an information, which is not contained within the context (i.e. the text) itself. It is legal however to say that the string of the context *occurs* in a foaf:Document. Additionally, str:Context is a subclass of str:String and therefore its instances denote Unicode text as well. The main benefit to limit the context is that an OWL reasoner can now infer that two contexts are the same, if they consist of the same string, because an inverse-functional data type property (str:isString) is used to attach the actual text to the context resource.

As a minor change, the human readable label will be dropped from the offset URI template, as it serves no function for machines. Here is the updated example again:

```
:offset_0_26546  a  str:Context ;
#the exact retrieval method is left underspecified
  str:occursIn <http://www.w3.org/DesignIssues/
       LinkedData.html> ;
# [...] are all 26547 characters as rdf:Literal
    str:isString   "[...]" .
:offset_717_729  a  str:String ;
    str:referenceContext :offset_0_26546 .
```

1.3 Additions for NERD

For NERD, three relevant concepts have to be covered by NIF: 1. OEN, 2. OED 3. NERD Ontology Types .

One Entity per Name. OEN can be modelled in a straightforward way, by introducing a property *sso:oen*[11], which connects the string with an arbitrary entity.

```
:offset_717_729  sso:oen dbpedia:Semantic_Web .
```

One Entity per Document. As documents are replaced by the notion of context, the property *sso:oec* is used to attach entities to a given context. We furthermore add the following DL-Axiom:

$$sso{:}oec \supseteq str{:}referenceContext^{-1} \circ sso{:}oen$$

As the property oen contains more specific information, oec can be inferred by the above role chain inclusion. In case the context is enlarged the information attached via the sso:oec property need to be migrated to the larger context.

[11]we use the prefix of the Structured Sentence Ontology (SSO) here, as this is the ontology modelling additional vocabularies of NIF (separation of concerns)

	|triples|	|tokens|	reasoning time	parsing time
no tokens	3	0	1.12s	0.01s
tokens	37234.88	8826.38	7.44s	0.38s
tokens (+String)	37482.20	8869.81	8.09s	0.38s
V1	24803.74	1709.69	5.38s	0.23s
V1 (+NERD)	26931.81	1709.69	7.39s	0.26s
V2	27951.35	1709.69	5.46s	0.27s
V2 (+NERD)	30079.42	1709.69	7.22s	0.29s
V3	22233.55	1709.69	4.63s	0.21s
V3 (+NIFNERD)	23395.84	1709.69	5.00s	0.23s

Table 1: NOTE: please ignore the POS part, it was done for something else. Measurement of the computational properties of different variants averaged over 100 random Wikipedia articles

NERD Ontology Types. The NERD Ontology types can not be assigned directly to NIF URIs as the potential referents (i.e. the universe of discourse) are disjoint. For example the NERD class nerd:Organization can only contain URIs, which refer to organizations[12]. The connection between NERD types and strings can be done in 3 ways, assuming:

```
# this URI points to the string "W3C"
:offset_23107_23110 a str:String;
  str:referenceContext :offset_0_26546 .
```

Variant 1 introduces a blank node and 2 triples:

```
# this URI points the the string "W3C"
:offset_23107_23110   sso:oen [nerd:Organization . ]
      .
```

Variant 2 uses an anonymous OWL class expression. We use DL notation for brevity as we would need 4 RDF triples:

$$\exists\ sso{:}oen.nerd{:}Organization\ (:offset\_23107\_23110)$$

Variant 3 we generate a new class for each NERD class by programmatically replacing `http://nerd.eurecom.fr/ontology#` by `http://nerd.eurecom.fr/nif#` and additionally migrate the subClassOf statements. The new classes are defined in analogy to this example:

$$nifnerd{:}Organization \sqsubseteq str{:}String$$
$$nifnerd{:}Organization \equiv \exists\ sso{:}oen.nerd{:}Organization$$

The actual assignment of the type to the string would only need one triple:

```
:offset_23107_23110  a  nifnerd:Organization .
```

Table 1 gives an overview over the computational advantages of Variant 3, which on a randomly selected Wikipedia article saves around 4144 triples and over 2 seconds time in combination with the simplified schema, when computing an inferred model[13]

2. REFERENCES

[12]Note that RDF assumes that all humans and all applications created by humans have perfectly agreed upon a globally defined meaning for each URI, thus resembling the hive mind of the Borg (`http://en.wikipedia.org/wiki/Borg_%28Star_Trek%29`). We also assume here that the reader knows exactly what is meant by "organizations".

[13]We used Pellet version TODO to materialize the inferred model