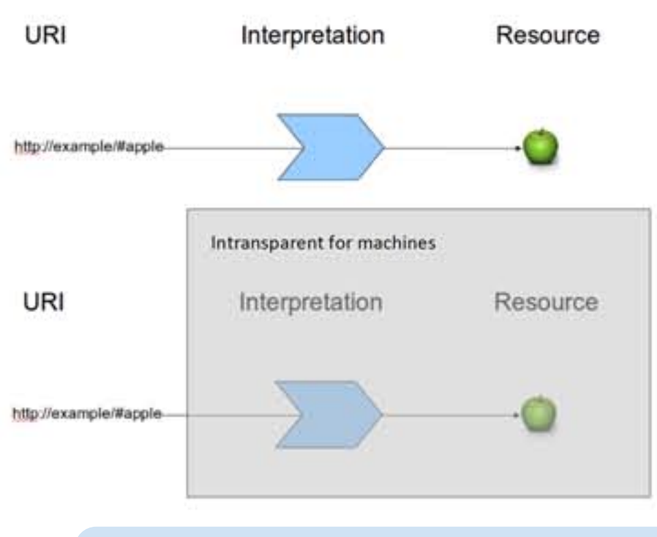
NLP Interchange Format (NIF)
A common data format for natural language processing (NLP)

Sebastian Hellmann
hellmann@informatik.uni-leipzig.de
Universität Leipzig, IFI/AKSW

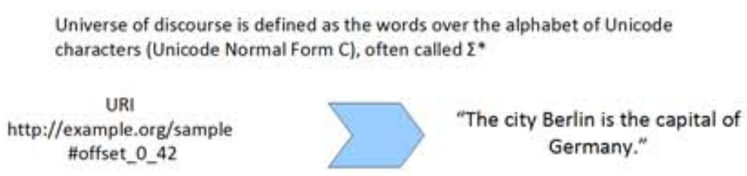
Jens Lehmann
lehmann@informatik.uni-leipzig.de
Universität Leipzig, IFI/AKSW

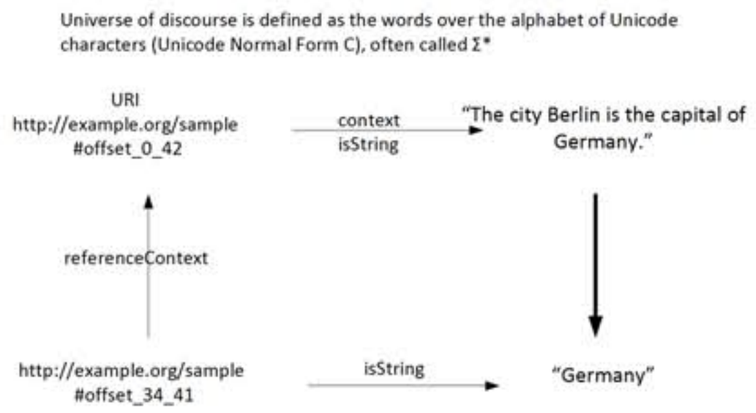
Sören Auer
auer@informatik.uni-leipzig.de
Universität Leipzig, IFI/AKSW

Martin Brümmer
bruemmer@informatik.uni-leipzig.de
Universität Leipzig, IFI/AKSW



http://nlp2rdf.org
http://aksw.org
http://lod2.eu





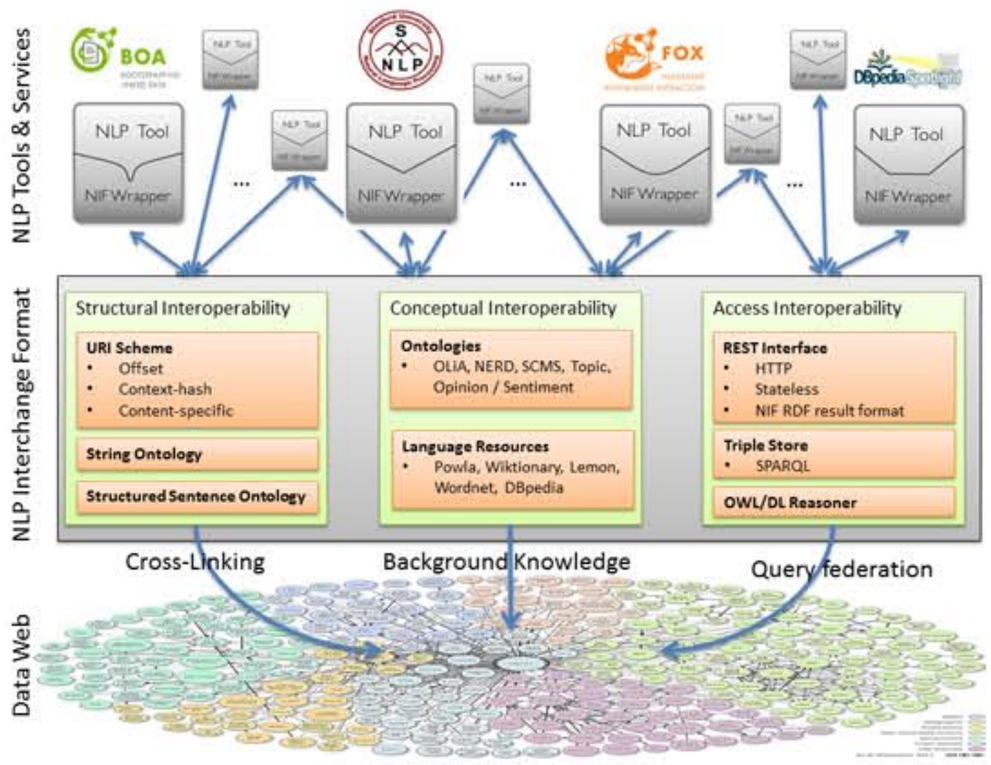
The NLP Interchange Format - NIF

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. An overview of the general NIF architecture is shown in the figure to the right. The core of NIF consists of a vocabulary, which allows to represent strings as RDF resources. A special URI design is used to pinpoint annotations to a part of a document. These URIs can then be used to attach arbitrary annotations to the respective character sequence. Employing these URIs, annotations can be published on the Web as Linked Data and interchanged between different NLP tools and applications.

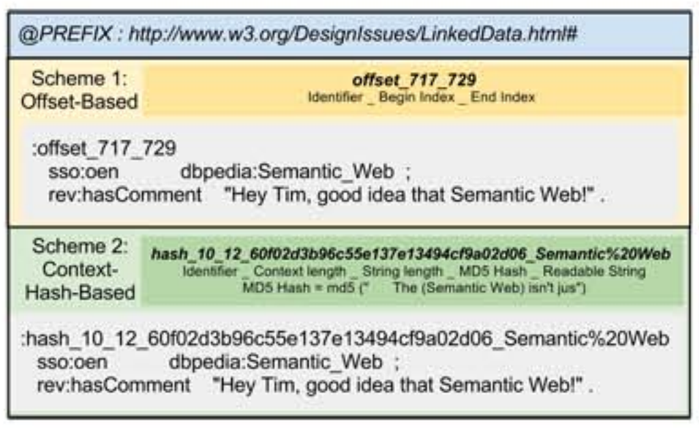
NIF addresses the interoperability problem on three layers: the structural, conceptual and access layer. NIF is based on a Linked Data enabled URI scheme for identifying elements in (hyper-)texts (structural layer) and a comprehensive ontology for describing common NLP terms and concepts (conceptual layer). NIF-aware applications will produce output adhering to the NIF ontology as REST services (access layer). Other than more centralized solutions such as UIMA and GATE NIF enables the creation of heterogeneous, distributed and loosely coupled NLP applications, which use the Web as an integration platform. Another benefit is, that a NIF wrapper has to be only created once for a particular tool, but enables the tool to interoperate with a potentially large number of other tools without additional adaptations. Ultimately, we envision an ecosystem of NLP tools and services to emerge using NIF for exchanging and integrating rich annotations.

NIF can be used for import and export of data from and to NLP tools. Therefore NIF enables to create adhoc workflows following a client-server model or the SOA principle. Following such an approach, clients are responsible for implementing the workflow. The client sends requests to the different tools either as text or RDF and then receives responses in RDF. This RDF can be aggregated into a local RDF model. Transparently, external data in RDF can also be requested and added without using additional formalisms. For acquiring and merging external data from knowledge bases, all existing RDF techniques and tools can be used.

The main platform for the adoption of NIF can be found at http://nlp2rdf.org. It serves as a host for NIF 1.0 and will host future specifications. Moreover, it builds a community around NIF by providing demos, development guides, code samples, tutorials, challenges and a mailing list.



NIF architecture aiming at establishing a distributed ecosystem of heterogeneous NLP tools and services by means of structural, conceptual and access interoperability employing background knowledge from the Web of Data.



NIF URI schemes: Offset (top) and context-hashs (bottom) are used to create identifiers

URI Design or How to address Strings with URIs?

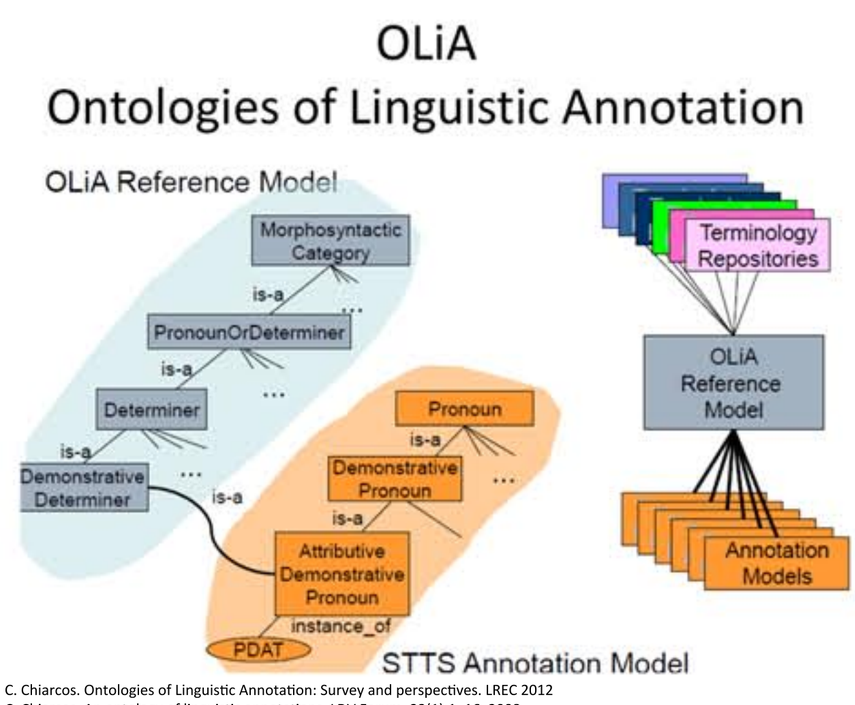
The fundamental rationale of NIF is to allow NLP tools to exchange annotations about documents in RDF. Hence, the main prerequisite is that parts of the documents (i.e. strings) are referenceable by URIs, so they can be used as subjects in RDF statements. We call an algorithm to create such identifiers *URI scheme*. For the URI creation scheme, there are three basic requirements - *uniqueness*, *ease of implementation* and *URI stability* during document changes. Since these three conflicting requirements can not be easily addressed by a single URI creation scheme, NIF 1.0 defines two URI schemes, which can be chosen depending on which requirement is more important in a certain usage scenario.
The offset-based URI scheme focuses on ease of implementation and is compatible with the position and range definition of RFC 5147 and builds upon it in terms of encoding and counting character positions. Offset-based URIs are constructed of four parts separated by an underscore '_': A *scheme identifier*, in this case the string 'offset'; the *start index* and the *end index*.
The context-hash-based URI scheme is designed to remain more robust regarding document changes. Context-hash-based URIs are constructed from five parts separated by an underscore '_': A *scheme identifier*, in this case the string 'hash'; the *context length* (number of characters to the left and right used in the message for the hash-digest); the *overall length* of the addressed string and the *message digest* (a 32-character hexadecimal MD5 hash created from the string and the context).
Since both URI schemes can be applied to arbitrary text documents, they are also applicable to HTML, XML, software source code, CSS etc. However, with its addressing scheme identifier NIF is extensible and further annotation schemes (e.g. more content-specific URI schemes such as XPath/XPointer for XML) can be easily included in future.

Structural and conceptual interoperablity via Ontologies

In addition to the URI scheme, structural interoperability in NIF is achieved using two ontologies: the String Ontology and the Structured Sentence Ontology (SSO).
The String Ontology comprises a class String and a property anchorOf to associate URIs in a given text and describe the relations between these string URIs.
The Structured Sentence Ontology (SSO) is built upon the String Ontology and provides additional classes for three basic units: sentences, phrases and words. Properties such as sso:nextWord and sso:previousWord can be used to express relations between these units. Furthermore properties are provided for the most common annotations such as the data type properties for stem, lemma, statistics, etc.

Conceptual interoperability is ensured in NIF by providing ontologies and vocabularies for representing the actual annotations in RDF. We divided the potentially different output of NIF tools into different NLP domains. For each domain a vocabulary was chosen that serves the most common use cases and facilitates interoperability. In simple cases, a property has been designated in the Structured Sentence Ontology. In the more complex cases fully developed linguistic ontologies which already existed were reused.

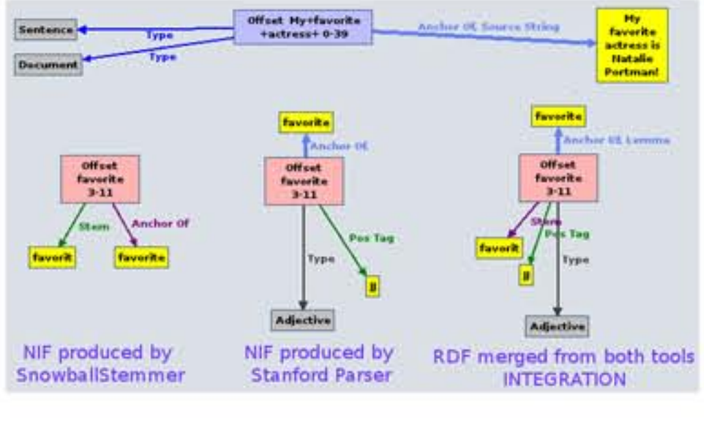
NIF OLiA provides a stable conceptual interface for applications. In the figure on the right we show how this interface is used. The annotations are provided by the Stanford POS Tagger, which uses the Penn Tag Set. OLiA provides an Annotation Model for the most frequently used tag sets, such as Penn. These annotation models are then linked to a reference model, which provides the interface for applications. Consequently, queries such as 'Return all Strings that are annotated (i.e. typed) as olia:PersonalPronoun' are possible, regardless of the underlying tag set. This also guarantees language independence.



C. Chiarcos. Ontologies of Linguistic Annotation: Survey and perspectives. LREC 2012
C. Chiarcos. An ontology of linguistic annotations. LDV Forum, 23(1):1–16, 2008.

A visualisation of NIF annotations and resulting merged RDF for the sentence 'My favorite actress is Natalie Portman!'.

The currently used RDF model might slightly differ from this image. It is for explanatory purposes only.



Learn more at http://nlp2rdf.org. Supported by http://lod2.eu.
Printed by Universitätsrechenzentrum Leipzig