Ontology-Based Interface Specifications for an NLP Pipeline Architecture

Ekaterina Buyko[1], **Christian Chiarcos**[2], **Antonio Pareja Lora**[3]

[1]Jena University Language & Information Engineering (JULIE) Lab, Jena, Germany
[2]University of Potsdam, SFB 632, Potsdam, Germany
[3]OEG, Universidad Politécnica de Madrid / DSIC, Universidad Complutense de Madrid, Madrid, Spain
[1]ekaterina.buyko@uni-jena.de [2]chiarcos@uni-potsdam.de [3]apareja@sip.ucm.es

Abstract

The high level of heterogeneity between linguistic annotations usually complicates the interoperability of processing modules within an NLP pipeline. In this paper, a framework for the interoperation of NLP components, based on a data-driven architecture, is presented. Here, ontologies of linguistic annotation are employed to provide a conceptual basis for the tag-set neutral processing of linguistic annotations. The framework proposed here is based on a set of structured OWL ontologies: a reference ontology, a set of annotation models which formalize different annotation schemes, and a declarative linking between these, specified separately. This modular architecture is particularly scalable and flexible as it allows for the integration of different reference ontologies of linguistic annotations in order to overcome the absence of a consensus for an ontology of linguistic terminology. Our proposal originates from three lines of research from different fields: research on annotation type systems in UIMA; the ontological architecture OLiA, originally developed for sustainable documentation and annotation-independent corpus browsing, and the ontologies of the OntoTag model, targeted towards the processing of linguistic annotations in Semantic Web applications. We describe how UIMA annotations can be backed up by ontological specifications of annotation schemes as in the OLiA model, and how these are linked to the OntoTag ontologies, which allow for further ontological processing.

1. Introduction

The maturation of language technology goes hand in hand with the creation of various corpora of linguistic annotations and libraries of NLP components, usually trained on these corpora. There is often a high level of heterogeneity between linguistic annotations, which ranges from the choice of arbitrary and idiosyncratic tags to fundamental differences in the conceptualisation of different linguistic categories and features, and over the degree of granularity of analyses. The Penn Treebank tag set, for example, distinguishes four tags for nouns, while the Susanne tagging scheme distinguishes 87 different tags for nouns. The divergence of linguistic tag sets usually complicates the interoperation of NLP components within a pipeline. For example, a parser trained on Penn Treebank requires input with the corresponding part of speech (POS) tags.

We focus here on the issue of the interoperation of NLP components in frameworks which are based on a data-driven architecture. The NLP components in a data-driven architecture do not share their code but only the processed data. Therefore, the interface specifications (input/output specifications) of single NLP components are crucial for building pipelines and exchanging various NLP components within these pipelines. Frameworks usually provide functionalities for defining and accessing such interface specifications. All components integrated in a framework are then characterized by abstract input/output specifications. The user usually has to define in advance what kind of data each integrated component may manipulate. This is achieved via the so-called *annotation type systems*.

Annotation type systems provide a format for the common representation of different tag sets and the technological infrastructure for the specification of a general taxonomy of linguistic concepts, relevant for linguistic annotations in an NLP pipeline. However, the annotation type systems in NLP frameworks usually do not aim to provide a conceptual basis for the linguistic annotations. This is where ontologies come into play. In the last years, several ontologies of linguistic terminology have been developed within different communities and for different purposes, e.g. OntoTag's ontologies (Aguado de Cea et al., 2004, focusing on NLP), GOLD (Farrar and Langendoen, 2003, focusing on language documentation), or the one proposed in Wilcock (2007, focusing on HPSG). So far, these ontologies of linguistic terminology have not converged into a single reference ontology of linguistic terminology and, in fact, it is more likely that community-specific ontologies will persist and be further developed according to the needs of their specific community.

In this paper, we suggest to link annotations to specific ontologies of linguistic terminology. We concentrate on the description of the mechanism by means of which such externally provided reference ontologies and concrete annotation types are linked, rather than proposing a direct generation of annotation types from one particular ontology. Especially, this modular approach allows to apply different reference ontologies in NLP pipelines, if required for a particular purpose.

The rest of the paper has been structured as follows: first, the outline of an ontological architecture for linguistic annotation, which re-uses some part of OntoTag's ontologies and incorporates it in an instantiation of OLiA, is presented in Sect. 2. Second, it will be shown how annotation type systems have been implemented in UIMA and formalized within the present proposal in Sect. 3. Third, some prospects and achievements are mentioned in Sect. 4. Fourth, some conclusions are outlined in Sect. 5.

2. An Ontological Architecture for Linguistic Annotation

In this section, we describe the approach followed and proposed in this paper, i.e. to link concrete annotations to a par-

ticular ontological representation, in order to achieve two specific goals:

(a) Abstraction from a particular tag set to ontological representations of linguistic annotations when processing this information.

(b) Application of ontology-based automatic reasoning to these annotations, based on the ontological representation of linguistic annotations.

Here, we concentrate on the first goal, namely, the description of the architecture which mediates between the annotations and a particular reference ontology. Issues concerning higher-level processing of linguistic annotations will be dealt with elsewhere.
In order to link annotations to ontologies of linguistic terminology, three components need to be distinguished, i.e.:

the reference ontology specifying the overarching terminological inventory into which concrete annotations originating from different tools are to be translated;

annotation schemes specifying the set of possible annotation values, their meaning and the restrictions on their interpretation within one particular type of annotation;

the mapping between concrete annotations and ontological concepts.

The reference ontology specifies the basic terminological inventory. In an NLP context with different tools trained on different tag sets, it may be interpreted as an 'interlingua' between different tag sets. However, as compared to existing pre-ontological accounts, e.g. Leech and Wilson (1996) and Atwell et al. (1994), a specific representation formalism is applied, which allows us to apply tools developed in the Semantic Web context. We re-use some part of the OntoTag ontologies as a reference ontology, as they are decisively targeted towards the integration of linguistic annotations and semantic reasoning in Semantic Web contexts. The OntoTag ontologies are described in Sect. 2.1.
As far as annotation schemes are concerned, explicit models of annotation schemes are usually represented by means of annotation documentation, e.g., tagging guidelines. It must be noted, however, that these guidelines are mostly intended for *manual* annotation and that tools trained on corpora tagged according to these guidelines may operate on substantial simplifications of these. Therefore, the mapping between annotations produced by some tool, and ontological concepts in the reference ontology requires a substantial degree of *interpretation* not only of the concrete tag (whose meaning may be insufficiently documented), but also of the terminological reference concept (that is language-independent). In existing approaches, e.g. in OntoTagger (Aguado de Cea et al., 2004), or in the one presented by Simons et al. (2004), this interpretation is represented implicitly in transformation scripts.
As for a mapping, however, these scripts are designed by programmers rather than linguists, and usually do not convey a discussion or justification for a particular mapping decision. In case a misinterpretation occurred,[1] it is very likely not to be recognized by the users of the ontological representation, and moreover, it may not be easily corrected, as the information about the mapping and the implementation are interwoven.
Therefore, we adopt an idea originally described by Chiarcos (2006) by applying a structured set of ontologies for linguistic annotations, in which the mapping between annotations and a reference ontology is expressed in a declarative way. Not only the reference ontology, but also the annotation schemes are modeled as ontologies in OWL/DL, and the linking between both ontologies is represented apart from either of these models by means of `rdf:descriptions`. This modular architecture of ontologies of linguistic annotations (OLiA) is described in Sect. 2.2.

2.1. OntoTag's Linguistic Ontologies

OntoTag is an abstract model created on purpose within the projects ContentWeb (Aguado de Cea et al., 2002) and PLAN-H-SemWeb (Aguado de Cea et al., 2004) for the hybrid (linguistic and ontological) annotation of Semantic Web (Berners-Lee and Fischetti, 1999) documents. OntoTag aims at describing the way in which multiple annotation tools can be integrated not only in a pipeline, but also in parallel, hence, enabling both interoperation and integration of NLP tools in an NLP architecture (Aguado de Cea et al., 2003).
The main components of OntoTag are: (a) its ABSTRACT ARCHITECTURE SPECIFICATION for NLP tool integration, which has to be instanced as a particular CONFIGURATION for each particular set of tools being integrated; and (b) its ABSTRACT ANNOTATION SCHEMA. A crucial resource underlying the model OntoTag and, more precisely, its abstract annotation schema, is its set of linguistic ontologies (Aguado de Cea et al., 2004), devised to represent the structure and relationships between elements of natural language at different linguistic levels (Aguado de Cea et al., 2002).
First of all, a LINGUISTIC LEVEL ONTOLOGY captures the stratification of natural language analysis and generation. Then, based on an extension of the EAGLES recommendations for morpho-syntactic and syntactic annotation (Leech and Wilson, 1996; Leech et al., 1996), three ontologies were implemented to represent category-attribute-value formalisms at all annotation levels (morpho-syntactic, syntactic, semantic, discourse and pragmatic): a LINGUISTIC UNIT ONTOLOGY, a LINGUISTIC ATTRIBUTE ONTOLOGY, and a LINGUISTIC VALUE ONTOLOGY. The Linguistic Unit Ontology includes all the units (categories) identified at different levels of annotation; the Linguistic Attribute Ontology includes the set of attributes associated to these units; and the Linguistic Value Ontology accounts for possible values of these attributes. Finally, a sort of upper-level ontology, the INTEGRATION ONTOL-

[1] Such misinterpretations occur regularly when informal abbreviations are used for tag names. For example, in the German tag set STTS, the tags for auxiliary verbs (`VA...`) are assigned to all forms of verbs which have forms that *can* serve as auxiliary verbs. Yet, the 'naive' interpretation of `VA...` tags is that it marks verbs that actually *act* as auxiliaries.

OGY, was built to link the rest of the ontologies in OntoTag, describing the relationships between the concepts in the other ontologies aforementioned.

OntoTag's ontologies were intended as a set of *heavyweight* ontologies, i.e., fully specifying the properties of each concept within the ontologies (e.g. the number of words in a *Multiword Token* or expression), enabling the insertion of instances for each concept (e.g. *15:00* as an instance of a *TIMEX Named Entity*), as well as detailing the axioms and rules related to each term in the ontology (e.g. which values are allowed for a given attribute in a given context) and subspecifying conveniently different relationships holding between the concepts in the ontologies (e.g. `SubClassOf`, `InstanceOf`, `Exhaustive`, `Disjoint`, `Partition`, `Part-Of`, etc.).

2.2. OLiA: Structured Ontologies for Linguistic Annotation

Coming to the description of the linking of OntoTag's ontologies with annotation type definitions for NLP, we adopt the structured model of ontologies of linguistic annotation (OLiA) as described in Chiarcos (to appear). Originally, this scenario was developed to enhance the access to heterogeneously annotated corpora, as part of the development of a sustainable archive of linguistic resources (Schmidt et al., 2006). In this context, an ontology was developed specifying reference concepts for ontology-based browsing and corpus querying (Rehm et al., 2008).

The core idea of the OLiA architecture, however, is a clear separation between the information drawn from the annotation documentation and its interpretation with respect to the reference terminology. This conceptual separation guarantees *transparency* and *sustainable maintenance* of the mapping between the annotations and the reference terminology.

For this purpose, it has been developed a structured, modular architecture, which allows for both the *lossless* ontological representation of specific annotations and their conceptual integration by reference to a general terminological backbone, termed REFERENCE MODEL in the OLiA architecture. OntoTag's ontologies represent fully-developed, heavyweight ontologies, making them the most interesting candidate for an external Reference Model for the purpose of NLP applications. Therefore, for the application described in this paper, the Reference Model is aligned with some part of the OntoTag ontologies, hence incorporating some of the knowledge captured by the OntoTag ontologies into OLiA.

Yet, not only an ontology conceptualizing reference concepts, but also ontologies formalizing different annotations schemes, so-called ANNOTATION MODELS, have been constructed. Annotation models are formalizations of annotation schemes which are exhaustive with respect to the annotation documentation available, but without any additional interpretation in terms of generally assumed linguistic categories, etc.

While an Annotation Model is specific for one particular language, community, or purpose, the Reference Model is a general terminological resource, and consequently based on a broad range of resources, including specific Annotation Models, grammatical references, textbooks, as well as existing terminological references such as the EAGLES recommendations for morpho-syntactic annotation (Leech and Wilson, 1996), and GOLD (Farrar and Langendoen, 2003). In case of divergent conceptualisations, e.g. the classification of attributive possessive pronouns as either Pronouns or Determiners, the EAGLES underlying taxonomy was taken as an orientation.

The Annotation Models and the Reference Model represent self-contained ontologies on their own. The conceptual integration of Annotation Models is then performed by means of a declarative LINKING between the Reference Model and each specific Annotation Model. In the linking, every concept (class) of the Annotation Model is assigned a superclass from the Reference Model, including complex superclasses composed with the set operators $\cup, \cap$, or $\setminus$, (see Rehm et al. (2008, Fig. 5) for an illustration). Users can verify the ontological interpretation of a particular annotation and, as the linking is specified apart from the Annotation Models and the Reference Model, it is even possible to modify the existing linking.

Also, researchers from different communities may not feel comfortable with specific design decisions and definitions adopted in the Reference Model and, consequently, the tripartite structure of Annotation Models, Reference Model, and the Linking between them can be augmented by the optional linking of the Reference Model with additional EXTERNAL REFERENCE MODELS, ontological formalizations of community- or language-specific terminological systems. Currently, we provide a linking with three external Reference Models, i.e., GOLD (Farrar and Langendoen, 2003), OntoTag's ontologies (Aguado de Cea et al., 2004), and an OWL representation of the specifications of the Data Category Registry (Ide and Romary, 2004).

Fig. 1 illustrates the resulting, integrative architecture comprising OntoTag as a reference ontology, some concrete annotations, and a set of structured ontologies mediating between both in an explicit, transparent and extensible way.

The upper part of the figure, related to OntoTag and the OLiA architecture, has been described in this section. The details of the integration of the ontology with UIMA annotation objects and their application to concrete annotations of a given piece of text is described in the following section.

3. Annotation Type Systems

In this section, it is described how annotations, formally represented by annotation objects – more particularly, by the UIMA annotation type system –, can be linked to the annotation models presented in the previous section.

3.1. UIMA Annotation Type Systems

For the research presented in this paper, the UIMA (Unstructured Information Management Architecture) Framework (Ferrucci and Lally, 2004) was selected as an example of a data-driven architecture. UIMA provides a platform for the integration of NLP components (ANALYSIS ENGINES in the UIMA) and the deployment of complex NLP pipelines. UIMA is a particularly suitable architecture for advanced text analysis applications such as text mining or information extraction.

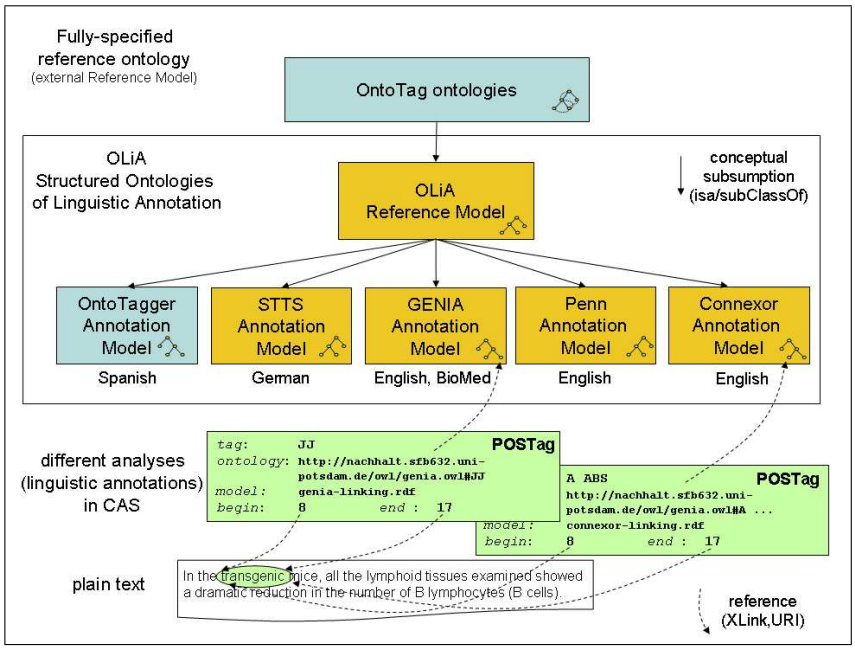

Figure 1: An integrative architecture linking annotations with full-fledged ontological representations via annotation types (CAS objects), OLiA Annotation Models and the OntoTag ontology.

The components in UIMA operate on common data by means of a construct referred to as CAS, COMMON ANALYSIS SYSTEM (Götz and Suhre, 2004). The CAS contains the subject of analysis (document) and provides meta-data in the form of annotations. Analysis engines receive annotations through a CAS and add new annotations to the CAS. An annotation in the CAS then associates meta-data with a region which the subject of analysis occupies (e.g., the start and end positions in a document). Annotations are thus *feature structures* associated with a region of the analysed text. CASes are crucial for the development and deployment of complex NLP pipelines.

All components integrated in UIMA are characterized by abstract input/output specifications. For the integration task, we define in advance what kind of data each component may manipulate. This is achieved via the UIMA *annotation type system*. In the type system, there are only two kinds of data, namely, types and features. *Features* specify slots within a type, which either have primitive values, such as integers or strings, or have references to instances of types in the CAS. *Types*, often called feature structures, are arranged in an inheritance hierarchy.

Several annotation type systems were developed within the UIMA framework. For example, Hahn et al. (2007) provides a domain-independent set of core annotation types at each linguistic annotation layer. These core types can be extended by domain-specific types. `POSTag` at the *Morpho-Syntax* layer is extended thus by types that represent various POS tag sets, e.g., `PennPOSTag` (Marcus et al., 1993) or `GeniaPOSTag` (Ohta et al., 2002). Hence, the definition of type hierarchies allows to refer from annotations to *types* rather than to specific string values (tags), providing a basis to solve the problems aforementioned, see Sect. 1.

However, a type system does not support any conceptual basis for the tags used in the annotations. Therefore, we propose here to keep a flatter hierarchy of type systems and to link the annotations to their conceptual definitions outside the UIMA type systems. In the following section, we provide a proposal for a linking of annotation types to their conceptual definitions.

3.2. Linking of UIMA Annotations to Ontological Definitions

In order to provide annotation types with a conceptual basis, we anchor their instantiations to specific ontologies. We propose to establish an appropriate linkage to the ontological resources via a clean interface definition (Fig. 1). The instantiations of the particular types of a type system are supplied with the features that link these instantiations to their conceptual definition in an ontological annotation model.

We exemplify here our scenario within the task of POS annotation in the UIMA framework. The type system contains the type POSTAG with its features (fields), e.g. *model* and *ontology*, that link a tag to its conceptual definition in the ontology (Fig. 2).

The *ontology* field conveys an OWL fragment specifying the ontological expression of a particular annotation within the current annotation model. The *model* field specifies an OWL file that imports the relevant OWL files (annotation models, OLiA reference model, external reference model, and the files specifying the linking between these).

The UIMA integrated POS tagger is assigned with a specification (descriptor) that provides the information about the annotation model and the ontology of POS tags provided by this particular POS tagger. With the help of this descriptor, the POS tagger can supply the annotations of the POSTAG

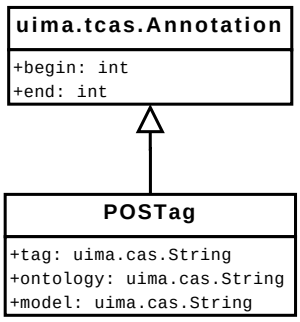

Figure 2: Class definition of the part-of-speech annotation type in a UIMA type system

objects with the information about their conceptual definition in the ontology.

As an example, the lower part of Fig. 1 shows a partial analysis for the sentence *In the transgenic mice, all the lymphoid tissues examined show a dramatic reduction in the number of B lymphocytes (B cells)*. Considering the token *transgenic*, different morphosyntactic analysers, here a tagger with the GENIA tag set and the Connexor parser, would produce different analysis represented in Fig. 1 by two alternative POSTAG objects. The tag assigned by the GENIA tagger is `JJ`. Consequently, the *ontology* field points to an individual `JJ` in the GENIA annotation model (slightly simplified in Fig. 1). Due to the information in the annotation model itself, and in the linking with the OLiA reference model and the OntoTag reference ontology, this information can be resolved to a particular set of triplets representing the linguistic information conveyed in this particular tag according to the ontologies. Figure 3 shows a screenshot from Protégé with several pieces of ontological information about the class `Adjective` in the GENIA annotation model. The individual `JJ` is a direct instance of `Adjective` and, thus, this information is inherited by `JJ` as well.

In the frame (subclass explorer), the class hierarchy of the GENIA annotation model (namespace `genia`) is shown, and so are the OLiA reference model (namespace `reference`) and the OntoTag ontologies as external reference model (namespaces `oio, luo, lao, lvo`). The upper part of the right frame (class editor) conveys a short description of the class `Adjective`. In the lower, right part of the class editor, the superclasses assigned to `genia:Adjective` and its particular linking with the concept `Adjective` in the reference model are shown. The $\equiv$ sign in front of the class name indicates equivalence with another concept, in this case the OntoTag concept `luo:Adjective`.

More important, however, is that the class inherits pieces of information from the reference model and from OntoTag, which is partly shown in the 'properties and restrictions' box in the central field of the class editor. So, from its annotation model, the class inherits the restriction that exactly one tag needs to be assigned (`genia:hasTag`, which is 'JJ' for the individual). Moreover, it inherits from the reference model the information that one or multiple values for case (`hasCase`), degree (`hasDegree`) are to be assigned, etc. Also, information from OntoTag is indirectly inherited via `reference:Adjective`, in particular the `M-S_Type` property that gives a processable characterization of the class according to OntoTag requirements. This information is essential when techniques for ontological reasoning that have been implemented for the OntoTag ontology are to be applied to annotations with the GENIA tag set.

The second POSTAG object in Fig. 1 gives a corresponding analysis. However, the ontological entry is more complex for this case, as two tags are assigned by Connexor, i.e. `A` meaning 'adjective' and `ABS` meaning that it occurs with positive (absolutive) degree. The ontological information that is abbreviated in the *ontology* field of the POSTAG object in the figure is more complex: it involves both the individual `connexor:A` and the property `connexor:hasDegreeOfComparison` linking it with the individual `connexor:ABS`, an instance of `connexor:Comparison` and `connexor:Feature`. The full OWL fragment of the *ontology* field is shown in Fig. 4.

One of the immediate prospects of the ontological representation now lies in the fact that this ontological specification, made in terms of two particular annotation models, can now be 'translated' into the corresponding ontological representations in the OLiA reference model and the OntoTag ontologies, thus allowing for a tagset independent representation of linguistic annotations. As such, the linking entails from the RDF description in Fig. 4 that the individual specified there is also an individual of the class described in Fig. 5, an OWL expression that means nothing but "adjective with positive degree".[2] On the basis of the linking between the OLiA reference model and the OntoTag ontologies, this expression may again be 'translated' into an expression according to the OntoTag ontologies. The following section describes how such abstractions may be applied within an NLP pipeline.

4. Prospects and Achievements

Anchoring the annotations to an existing ontology represents a methodological advantage per se, especially, as it allows to base interface specifications on terminological resources developed – and evaluated – by a particular community, rather than developing an independent taxonomy of abstract annotation types from scratch. Beyond this, fully specified ontologies also allow to reason over annotations, especially in the cases of *integration* and of *comparison* different annotations, natural handling of *underspecification*, and mechanisms for fine-grained *information reduction*. We illustrate the implementation of these reasoning tasks in OntoTagger, a particular instantiation of OntoTag's Abstract Architecture Specification (Arrizabalaga-Hernández, 2004; Serradilla-Fernández, 2004) in Sect. 4.1. We show how this methodology can be extended and applied to the *(heuristic) translation* between different tag sets in Sect. 4.1.

[2] Note that fundamentally different conceptualizations may be bridged when linking reference model and annotation models. For the sake of space, we chose a trivial example as illustration.

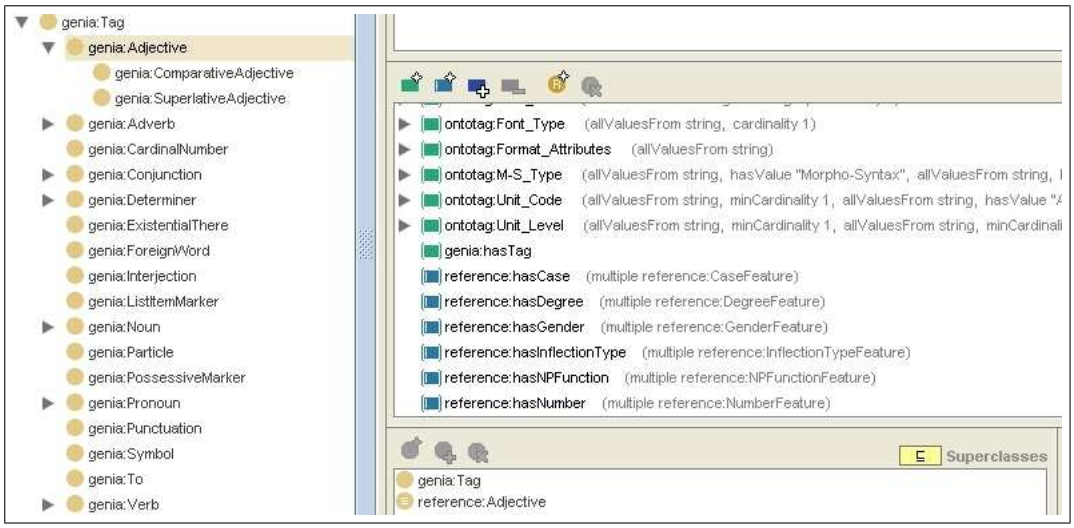

Figure 3: Screenshot from Protégé showing the class `Adjective` in the GENIA annotation model

```
<rdf:description rdf:about="http://nachhalt.sfb632.uni-potsdam.de/owl/connexor.owl#A">
  <connexor:hasDegreeOfComparison rdf:resource="http://nachhalt.sfb632.uni-potsdam.de/owl/connexor.owl#ABS"/>
</rdf:description>
```

Figure 4: Example of a comlex OWL fragment in the *ontology* field entry of a POSTAG annotation object.

4.1. Integration and Comparison of Multiple Annotations

Besides the sheer integration of different annotations within the OLiA reference model, the use of fully specified ontologies also provides some general benefits, such as enabling the enrichment and the underspecified representation of linguistic analyses. Such studies have been performed in the Semantic Web context using the OntoTag ontologies, and due to the linking of the OntoTag ontologies with the annotation models and UIMA annotation types, these results and the methodology employed may be directly applied to annotations in an UIMA pipeline, as well. Here, some key results and experiences using the OntoTag ontologies are shortly summarized.

In OntoTagger, the use of conceptual definitions is a crucial factor for the integration and comparison of the annotations generated by the different tools integrated in the configuration operating at the morphosyntactic level. Indeed, there are several tools generating annotations at this level. Since a unique and unambiguous POS tag (whenever possible) has to be produced as output by the combination of these annotations: (i) they have to be compared in order to determine which ones are correct; and (ii) it has to be identified which one(s) of these correct tags is (are) the most accurate of them.

Considering that each tool has its particular tagset, and that these tagsets do not match trivially, a higher (conceptual) level has been established on top of them, so that each one can be mapped and refer to it. A tailored annotation mapping module (the so-called STANDARDISATION PHASE) has been developed for each tool, which performs the correspondence between the tags coming from the tool and the related concept(s), attributes and values within OntoTag's ontologies, based on a fashion of XML mapping files (comparable to the linking between annotation models and the reference model, but based on a task-specific XML representation).

For example, one of the tools might annotate the Spanish word *pasado* in the context *el pasado festival de Sitges* (the last Sitges festival) with an `AJMS` POS tag (adjective, masculine, singular); another one with a `Nombre Común` (common noun) POS tag plus a numerical code identifying its morphological analysis (masculine, singular); and another one with a disambiguated set of tags which includes the previous ones, but coded according to its particular numerical jargon. Then, in order to determine which one of the POS analyses is the most accurate for the given context, their associated concepts in the OntoTag ontologies are retrieved by means of the mapping module and, then, compared at this conceptual level. Then, some contextual heuristics are applied to disambiguate the analysis at this same conceptual level, to determine the correct analysis, and to produce an output in terms of the associated ontological concepts (such as `oio:Linguistic_Unit`, `luo:Common_Noun`, `oio:Linguistic_Attribute`, `lao:Gender`, `oio:Linguistic_Value`, or `lvo:Masculine`).

The case of handling underspecification is rather equivalent: for example, one of the tools might annotate the (Spanish) word *formidable* in the context *esta formidable adaptación* (this formidable adaptation) with an `AJ` POS tag (adjective - no morphological information); another one with an `Adjetivo` (adjective) POS tag plus a numerical code identifying a disambiguated morphological analysis (masculine or feminine, singular); and another one with a tag which includes the same information as the previous ones, but coded according to its particular numerical jar-

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <rdf:description rdf:about="http://nachhalt.sfb632.uni-potsdam.de/owl/e-eagles.owl#Adjective"/>
    <owl:Class>
      <owl:Restriction>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:about="http://nachhalt.sfb632.uni-potsdam.de/owl/e-eagles.owl#hasDegree"/>
          <owl:allValuesFrom rdf:resource="http://nachhalt.sfb632.uni-potsdam.de/owl/e-eagles.owl#Positive"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```

Figure 5: Class description of the individual described in Fig. 4 in terms of the OLiA reference model.

gon. With a similar process, using the conceptual representation formalized in the ontologies of OntoTag, it can be detected that the first tag is an underspecification of the other two and, using the linguistic context, it is determined that these other two are an underspecification of the most accurate tag in that context, which is the one eventually assigned, and which corresponds to a feminine singular adjective (expressed in terms of OntoTag's ontologies).

Thus, the application of OntoTag's ontologies for the integration and validation of different NLP tools has already been experimented successfully. However, integrating this approach within an extensible, modular architecture requires an easier way to represent and modify the rules for the mapping of annotations to ontological concepts; it is here where the ontological architecture presented in this paper is assumed to be quite useful.

Consequently, an ontology-based UIMA-integrated NLP pipeline can be designed along similar lines, with robust ontological representations for different linguistic annotations achieved by the integration of a multitude of taggers. Furthermore, ontology-based interface specifications provide a natural abstraction over tags using their conceptual definition, and reduce thus the amount of morphosyntactic information. For example, for some components such as a Named Entity Tagger, the morphosyntatic information could be reduced to reference categories such as *Adjective* or *Noun*.

Therefore, the perspective field of application for the architecture described in this paper lies in pipelines of NLP modules which directly take ontological descriptions as their input. This type of pipelines are explored in the following section.

4.2. Translating between Annotations

For the application within a NLP pipeline of *existing* modules operating on concrete annotations rather than ontological specifications, the reference ontologies may be exploited to construct a heuristic, but transparent mapping between different annotations. As a result, NLP modules trained on different annotation schemes can be combined with each other.

Hence, a tag a from tag set A is assigned an ontological representation c in terms of the reference ontology. Given that c is the most specific OntoTag concept which subsumes a, then all instances of c in tag set B are consulted, and some tag (individual) b which is most precisely rendered by c (but not by any other concept in the reference ontology) can be assumed as the tag corresponding to a. In case there is another tag b' which is also an instance of c in B, then the most frequent candidate is heuristically determined as the probable counterpart of a.

Applied to the example in Fig. 1, this allows to translate between GENIA and Connexor annotations. As such, the GENIA POSTAG object refers to the individual `genia:JJ`, which is an instance of `reference:Adjective`. Moreover, it is also unmarked with respect to comparison (while the tags `JJR` and `JJS` are comparative and superlative adjectives, respectively), and thus, interpreted as a positive adjective in the linking. Therefore, this tag is translated into an OLiA reference model representation which is *identical* to the one shown in Fig. 5. Reverting the construction of this presentation from the corresponding Connexor tag which was described above yields the corresponding Connexor tag. As a result, taggers producing GENIA-conformant annotations may be combined with NLP modules expecting Connexor-type annotations.

5. Conclusions and Outlook

In this paper, we have described the linking between ontological annotation models and annotation instantiations in a data-driven architecture. This task is described with special consideration of the UIMA framework and the OntoTag ontologies. Yet, as multiple ontologies of linguistic terminology have been designed within different communities and for different purposes, we employ an architecture that allows to be adapted to different reference ontologies: a set of structured ontologies mediate between a reference ontology and concise annotation types. In the NLP pipeline, the linkage between annotations and their conceptual definitions can be then exploited by modules in order to integrate and convert existing annotations.

More generally, ontology-based specifications of linguistic annotations follow a general trend for convergency in semantic and grammatical processing and annotation formats established in the last years, cf. Cimiano and Reyle (2003), Ben-Avi and Francez (2004), Blythe and Gil (2004), Hovy et al. (2006), and Burchardt et al. (2008). The linking between concrete annotations in an NLP pipeline and ontologies of linguistic annotations is to be seen as another step in this development.

In the long run, the integration of ontologies of linguistic annotation with UIMA annotation type specifications is to be seen as a necessary pre-condition for the development of NLP pipelines operating on robust, tool-independent onto-

logical specifications, whereas currently applied processing modules are annotation-specific and thus restricted in their potential combination within an NLP pipeline.

6. Acknowledgements

The research described in this paper has been partially funded within the BOOTStrep project under grant FP6-028099, the project "Sustainability of Linguistic Data" (SFB 441/C2, funded by DFG), and GeoBuddies (Spanish Ministry of Science and Technology, TSI2007-65677C02).

7. References

G. Aguado de Cea, I. Álvarez de Mon y Rego, A. Gómez-Pérez, A. Pareja-Lora, and R. Plaza-Arteche. 2002. A semantic web page linguistic annotation model. In *Proc. AAAI'2002 Workshop: Semantic Web Meets Language Resources (AAAI Technical Report WS-02-16)*, pages 20–29.

G. Aguado de Cea, I. Álvarez de Mon y Rego, A. Gómez-Pérez, and A. Pareja-Lora. 2003. OntoTag: XML/RDF(S)/OWL Semantic Web page annotation in ContentWeb. In *Proc. NLPXML-2003*.

G. Aguado de Cea, I. Álvarez de Mon y Rego, and A. Pareja-Lora. 2004. OntoTag's linguistic ontologies: Enhancing higher level and Semantic Web annotations. In *Proc. LREC 2004*, volume VI, pages 1905–1908.

F. J. Arrizabalaga-Hernández. 2004. OntoTagger: Herramienta de anotación lingüístico–ontológica. M.Sc. Thesis, Universidad Politécnica de Madrid.

J. Atwell, , J. Hughes, and C. Couter. 1994. AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models. In Resnik P Klavans J, editor, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*. Association for Computational Linguistics, Las Cruces.

G. Ben-Avi and N. Francez. 2004. Categorial grammar with ontology-refined types. In *Proc. Categorial Grammars – An efficient tool for Natural Language Processing*, Montpellier, France.

T. Berners-Lee and M. Fischetti. 1999. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper, San Francisco.

J. Blythe and Y. Gil. 2004. Incremental formalization of document annotations through ontology-based paraphrasing. In *WWW '04: Proc. 13th Int'l Conf on World Wide Web*, pages 455–461, New York, NY, USA. ACM.

A. Burchardt, S. Pado, D. Spohr, A. Frank, and U. Heid. 2008. Formalising Multi-layer Corpora in OWL DL - Lexicon Modelling, Querying and Consistency Control. In *Proc. 3rd Int'l Joint Conf on NLP (IJCNLP 2008)*, Hyderabad, India.

Ch. Chiarcos. 2006. Avoiding Data Graveyards: Deriving an Ontology for Accessing Heterogeneous Data Collections. In *Proc. Ontologies in Text Technology (OTT'06)*, pages 113–118, Osnabrück, Germany.

Ch. Chiarcos. to appear. An ontology of linguistic annotations. *GLDV-Journal for Computational Linguistics and Language Technology*.

P. Cimiano and U. Reyle. 2003. Ontology-based semantic construction, underspecification and disambiguation. In *Proc. Prospects and Advances in the Syntax-Semantic Interface*.

S. Farrar and D. T. Langendoen. 2003. A Linguistic Ontology for the Semantic Web. *GLOT International*, 7:97–100.

D. Ferrucci and A. Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

T. Götz and O. Suhre. 2004. Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal*, 43(3):476–489.

U. Hahn, E. Buyko, K. Tomanek, S. Piao, J. McNaught, Y. Tsuruoka, and S. Ananiadou. 2007. An annotation type system for a data-driven NLP pipeline. In *Proc. Linguistic Annotation Workshop (LAW 2007)*.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2006. OntoNotes: The 90% Solution. In *HLT-NAACL06*.

N. Ide and L. Romary. 2004. A registry of standard data categories for linguistic annotation. In *Proceedings of the Fourth International Language Resources and Evaluation Conference (LREC)*, pages 135–139, Lisbon.

G. Leech and A. Wilson. 1996. Recommendations for the morphosyntactic annotation of corpora (EAG–TCWG-MAC/R). Technical report, EAGLES.

G. Leech, R. Barnett, and P. Kahrel. 1996. Recommendations for the Syntactic Annotation of Corpora (EAG–TCWG-SASG/1.8). Technical report, EAGLES.

M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The PENN TREEBANK. *Computational Linguistics*, 19(2):313–330.

T. Ohta, Y. Tateisi, and J.-D. Kim. 2002. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In *Proc. of the 2nd HLT*, pages 82–86.

G. Rehm, R. Eckart, Ch. Chiarcos, and J. Dellert. 2008. Ontology-Based XQuery'ing of XML-Encoded Language Resources on Multiple Annotation Layers. In *Proc. LREC 2008*, Marrakech, Morocco.

Th. Schmidt, Ch. Chiarcos, T. Lehmberg, G. Rehm, A. Witt, and E. Hinrichs. 2006. Avoiding Data Graveyards: From Heterogeneous Data Collected in Multiple Research Projects to Sustainable Linguistic Resources. In *Proc. E-MELD Workshop on Digital Language Documentation*, East Lansing, Michigan.

A. I. Serradilla-Fernández. 2004. Integración de recursos semánticos y aprendizaje de named entities en OntoTagger. M.Sc. Thesis, Universidad Politécnica de Madrid.

G. F. Simons, W. D. Lewis, S. O. Farrar, D. T. Langendoen, B. Fitzsimons, and H. Gonzalez. 2004. The semantics of markup: Mapping legacy markup schemas to a common semantics. In *Proc. NLPXML-2004*, pages 25–32, Barcelona, Spain.

G. Wilcock. 2007. An OWL Ontology for HPSG. In *Proc. ACL 2007*, pages 169–172, Prague, Czech Republic.