Modeling, Building and Maintaining Lexica for Corpus Linguistic Studies

Rüdiger Gleim[1], Armin Hoenen[1], Nils Diewald[2], Alexander Mehler[1], Alexandra Ernst[1]

[1]Goethe-University Frankfurt, [2]Bielefeld University

1 Introduction

Many corpus linguistic studies depend on powerful lexical resources to support automatic lemmatization, part-of-speech-tagging and related tasks of preprocessing corpus data. This is all the more relevant when long-term historical corpora are analyzed, because the underlying language changes over time, which creates additional challenges. An example of such a corpus is the Patrologia Latina (Migne, 1865) that includes documents from the 4^{th} to the beginning of the 13^{th} century. It unfolds several stages of the development of Late Latin in the direction of Early Romance on various levels of linguistic resolution (Clackson and Horrocks, 2007). Another example of demanding corpora are the complete works of novelists. To adequately preprocess such texts it is important to reflect the use of lexicographical variations specific to the author. Spelling errors for example should not be normalized but properly handled, for example by annotating and including them into the lexicon.

Creating and maintaining adequate lexical resources to preprocess and analyze such corpora are challenging tasks which relates to *linguistic, informational* and *methodological* aspects. From a linguistic perspective, the usage of lexical resources for corpus linguistic studies hinges upon the availability of an underlying data model that is flexible enough in several respects: first of all, the model must reflect morpho-syntactic specifics of a range of *different* languages. Then, the model has to account for derivations, compositions and other word relations. This is required, among other things, for multi-word units up to the level of collocations as considered in corpus linguistic studies. Furthermore, the data model should account for the change of the expression and content plane of lexical units as well as for the change of their grammatical usage in order to reflect the temporal dynamics of the lexicon.

These linguistic requirements give an impression of the complexity the data model has to deal with from an informational perspective. The typology of entities and relations can hardly be cast into a fixed form but need to be captured by an extensible design. Finally, this complexity has to be kept manageable by offering a proper interface and by supporting interoperability with established formats (e.g., TEI P5 TEI Consortium (2010) or RDF W3C (2010)).

We chose to tackle the problem by starting with an abstract definition of the data model, the *eLexicon Data Model*. On this basis, following the well-known three layer approach according to the ANSI/X3/SPARC Study Group on Data Base Management Systems (ANSI, 1975), we then define increasingly concrete data models. Separating these levels of abstraction help to keep technical issues of implementation apart from the general design of the lexicon. Since the appropriateness and efficiency of a concrete data model depends on its application we present two different approaches which rely on the same basis. For operational use we propose a logical data model which is based on the paradigm of normalized relational data modeling. For interoperability we have implemented an RDF based data model which can also be used in specialized RDF Database Systems. In addition to interoperability, RDF provides better means for validation of lexicon instances. The focus of this article lies in the description of these two logical data models and their application. To complete the technological part we have developed the *eLexicon Browser* as part of the eHumanities Desktop (Gleim and Mehler, 2010). Using a web interface, it mediates between the complexity of the data model and its usability.

The paper is structured in two parts. In the first part, section 2 introduces the data models as well as means of searching and browsing lexical resources. More specifically, we start by describing the general concepts of the eLexicon Data Model in section 2.1. On this basis we then describe our two logical data models, namely the relational approach in section 2.2 and the RDF based design in section 2.3. Section 2.4 describes the eLexicon Browser which concludes the technical part of the paper. In the second part we exemplify the use of the eLexicon by three lexicon instances. Section 3.1 describes works on a lexicon of Late Latin. Section 3.2 introduces a lexicon of Avestan and section 3.3 concludes with a lexicon which reflects the language used in the works of Hugo von Hofmannsthal.

Finally we give a conclusion and prospect of future work.

2 Modeling and accessing lexical resources in the eLexicon

The main task from an informational perspective is how to accurately model the entities of a lexicon and their relations. On the one hand, the model has to be highly adaptable to cover the diversity of different languages and variations thereof. Ideally it should be extensible, so that when unpredicted requirements arise, the existing model will not have to be changed. On the other hand it should be constrained to a certain degree so that it does not shift into arbitrariness. And last but not least we need appropriate means for persistance and adhere to established standards. In order to disentangle the complexity we apply the well-known three-layer model of data modeling according to ANSI/X3/SPARC Study Group on Data Base Management Systems (ANSI, 1975). It distinguishes between conceptual, logical and physical data model instances which build upon one another. With each step the data model shifts from an abstract to more concrete and technical state. We start on the conceptional level by describing the eLexicon Data Model. Subsequently we describe two logical model instances and conclude with the graphical user interface, the eLexicon Browser.

2.1 eLexicon Data Model

Transfered to the problem at hand, the conceptual data model should give an abstract description of the entities, attributes and relations of a lexicon. This perspective helps to focus on the requirements of the data model without having to keep in mind a paradigm for technical implementation. Figure 1 sketches our conceptual data model for lexicons, the *eLexicon Data Model*. In this article we keep the description of the eLexicon Data Model brief and instead focus on the logical and physical level which are more relevant for building and accessing lexicon instances.

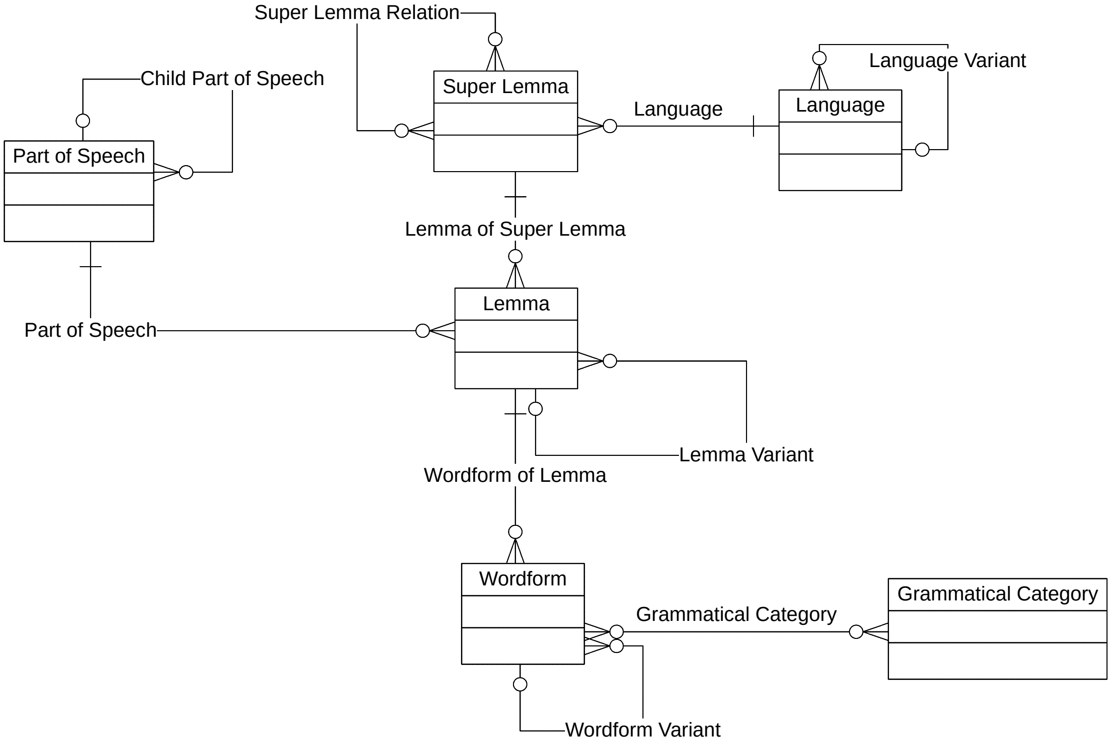


Figure 1: Simplified ER Diagram of the eLexicon Data Model

We basically distinguish between lemmas and word forms. A word form is bound to one lemma and defined by a set of relations which cover the grammatical categories such as tense, case, mood and so on. Both lemma and word form can relate to variants. We distinguish variants for both because variants on the lemma level systematically affect their inflection. A trivial example would be variants in latin orthography regarding the usage of “u” and “v”. Variants on word form level on the other hand are local and do not have an effect on other inflectional paradigms of the underlying lemma. “Variation” is given as one example of the multitude of relations lemmas and word forms can have. Not displayed but accounted for are, among others, semantic relations and chronological ordering. A given lemma is related to a specific part of speech. Parts of speech may have children to allow for a tree strucure. A set of lemmas which are, for example, variants of one another but share the same meaning are grouped together by a “Super Lemma”. A Super Lemma belongs to a specific language. The latter can, similar to the parts of speech, form a tree structure.

2.2 Relational Data Model

The conceptual data model gives a rather abstract idea of the lexicon’s entities and their relations. For productive use a logical and physical data model instance is needed. In the following we describe an approach which is based upon the relational database paradigm. It is an adaption of a data model used to model semantic databases (Mehler et al., 2008). The ER diagram in figure 2 gives an overview of the relational database model. *Objects* serve the representation of both lexicon data as well as types. That is, superlemmas, lemmas, word forms and their types are all modeled as objects. Using the *has*

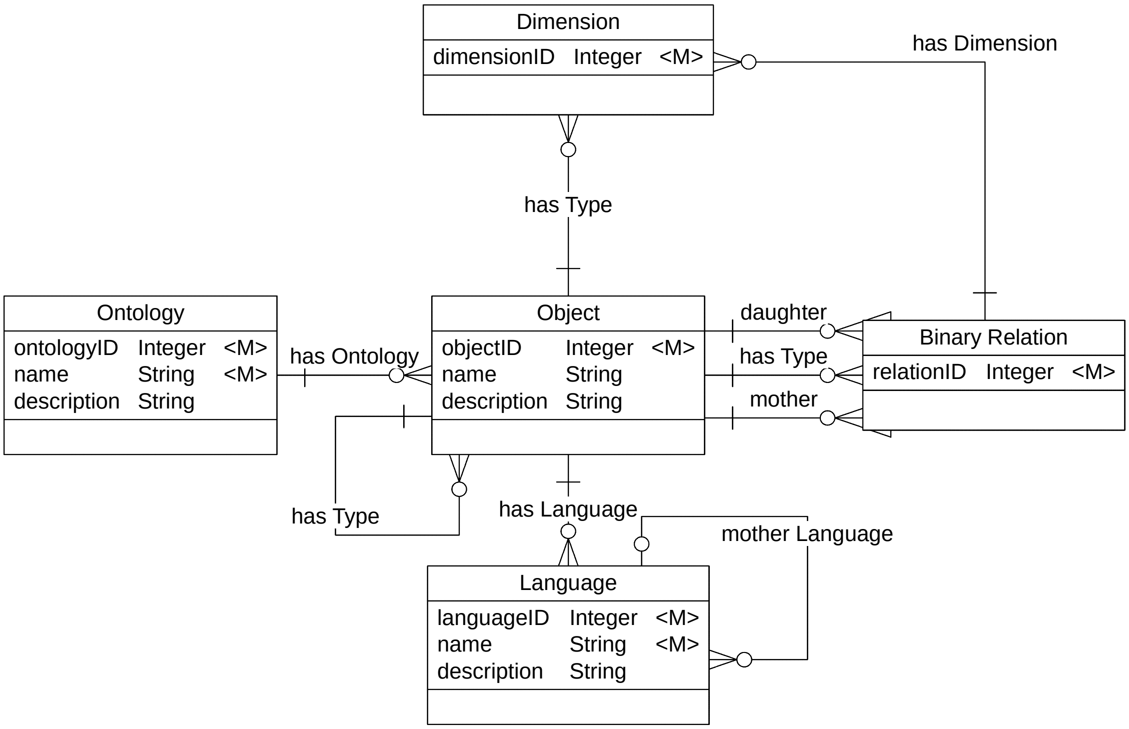


Figure 2: E/R Diagram of the Relational Data Model

Type relation, types can form trees and be assigned to a specific ontology. Lexicon instances use types to model part of speech trees, types of semantic relations and so on. Objects must be assigned to a specific language. This allows for lexicon entries of different languages in one lexicon instance. Languages themselves can be structured as trees to model language families or variations thereof. *Binary Relations* are used to model any kind of relation between lexicon entries, as for example that a word form is an inflected form of a specific lemma. Following this example there is the need to specify the grammatical categories of such a “lemma to word form” relation. What is the mood, case, etc.? We model this as multiple typing of binary relations. This is achieved by assigning an arbitrary number of *dimensions* to a binary relation. Dimensions themselves are typed according to an ontology stored as objects.

Figure 3 illustrates how a lexicon instance can use objects to build an ontology of types. The example shows that all types have a common root. On this basis (among others) grammatical categories, parts of speech, relation types, as well as types of lexical entries are distinguished. So when descending to the leaves the typing gets increasingly specific until, for example the grammatical case *Accusative* is defined as type.

How can such an ontology be used when representing lexical entries and relations? Figure 4 exemplifies this by instance of the lemma “Caesar”. The figure is divided into two parts. The upper part shows objects which represent lexical entries. The lower part contains objects which relate to the ontology of types. Keep in mind that all objects share the same domain. The object which represents the lemma “Caesar” has the objectID 100 and is typed as “Person”. “Person” is a special case of the part

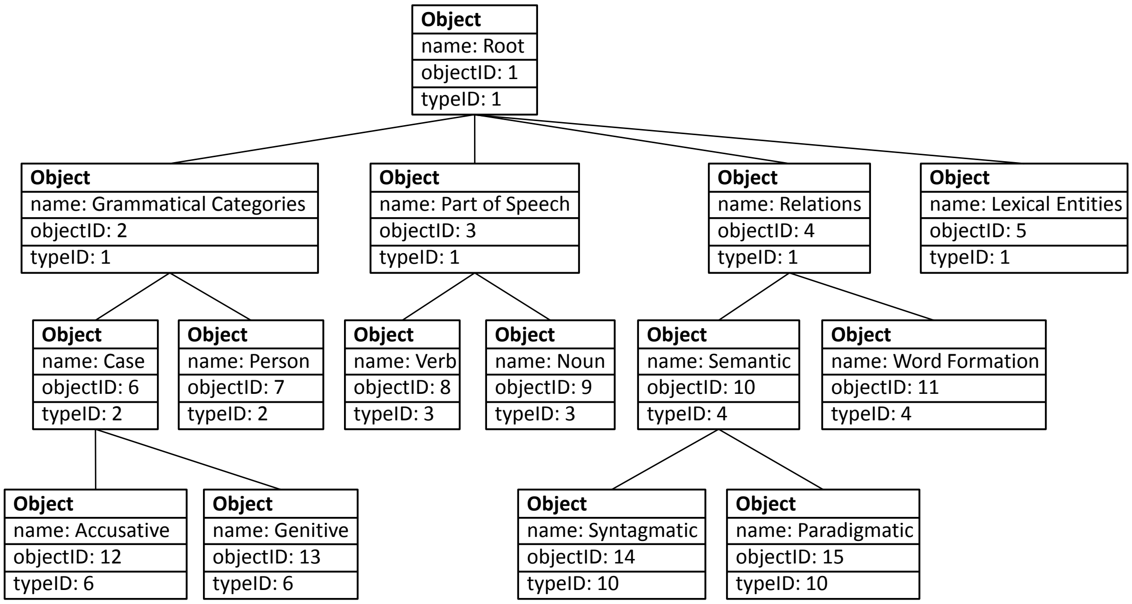


Figure 3: Example of how the relational model can be used to represent a type system.

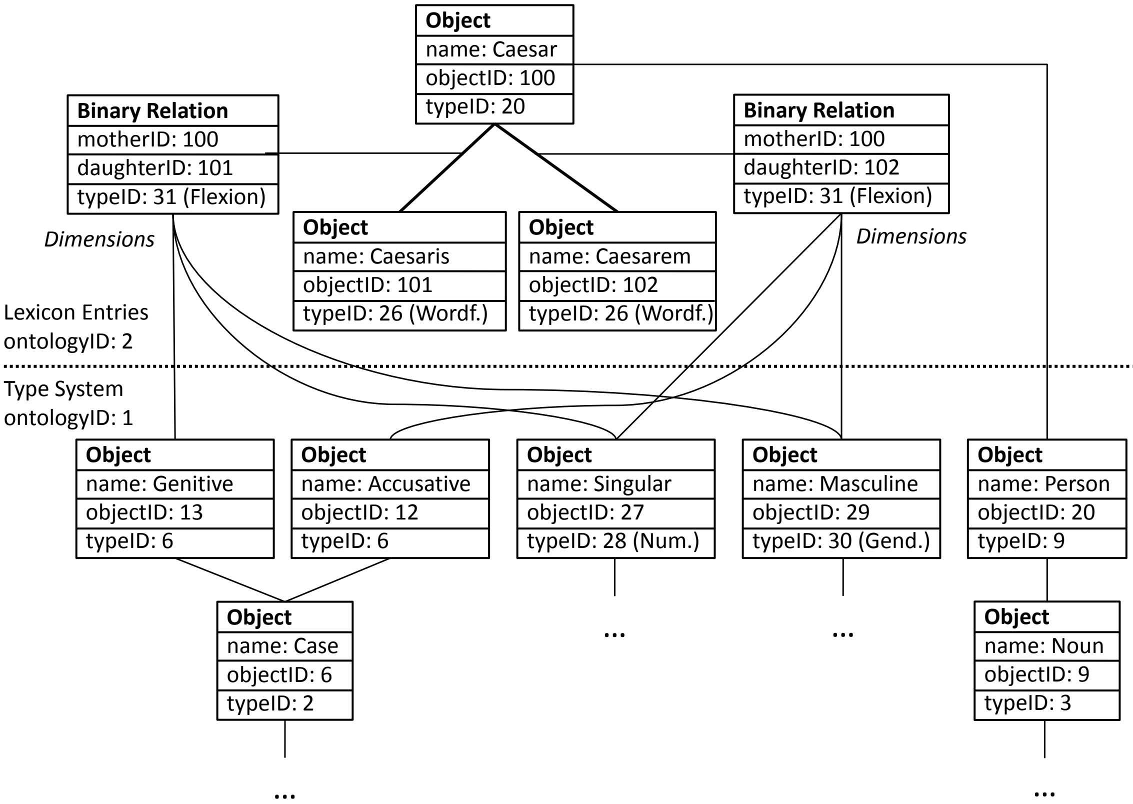


Figure 4: Sample representation of a lemma and two flexions thereof.

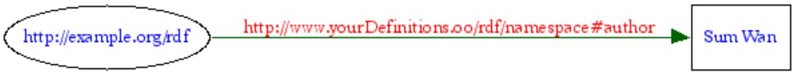
of speech “Noun”. This example shows two inflections of the lemma: “Caesaris” as genitive, singular, masculine and “Caesarem” as accusative, singular, masculine. The fact that those word forms are flexions of the lemma “Caesar” is represented by a binary relation. The relation targeting “Caesaris” is typed multiply by dimensions which are of the desired types “Genitive”, “Accusative” and “Singular”. The other word form is represented likewise. More complex relations such as derivation, semantic relations etc. are modeled in the same manner.

This relational data model can easily be transferred into a physical data model instance for a relational database management system such as MySQL[1]. We currently use the relational approach in our productive environment that is, lexicon instances are created as relational databases which can be accessed via the eLexicon Browser (see section 2.4).

2.3 RDF Data Model

In this section, prior to the explanation of our concrete RDF schema, the general RDF data model will be introduced. RDF serves as a logical layer of our three-layer approach following the ANSI/X3/SPARC model, just as does the relational database model. Then a sketch of the RDF-XML outlines the general shape of our physical layer. The last paragraph finally describes the specific characteristics of our implementation in more detail.[2] The Resource Description Framework(RDF) is a data model issued by the W3C in 1999 (Hitzler et al., 2008). As an ontology language it constitutes the so called “semantic web” by virtue of adding labeled metadata and relations among them to web resources via RDF *triples*. An RDF triple is a relation between three entities, one *subject*, one *predicate* and one *object*, a terminology that should not be confused with the linguistic counterparts of these labels. The RDF subject is the entity that is specified by the triple, the predicate denotes the kind of relation that exists between a subject and an object and the object is the entity to which the subject is related. Only the object can be a so called *literal* (that is a [typed] value), subject and predicate must have URIs. This architecture was originally designed to describe and relate web resources.

subject: http://example.org/rdf predicate: author object: Sum Wan



Listing 1: A basic RDF example.

```
<RDF>
  <Description about="http://example.org/rdf">
    <author>Sum Wan</author>
  </Description>
</RDF>
```

Since the beginning of RDF in 1999 the number of triples on the web has been rising exponentially. Starting with a few triples in the first years of the young millennium, by September 2010 the number had already risen to 19.5 billion (Kenekayoro, 2011). Global players such as NASA and the NY

[1] http://dev.mysql.com

[2] Terminology in this section might stem from the rdf-syntax http://www.w3.org/1999/02/22-rdf-syntax-ns or RDFS-schema file http://www.w3.org/2000/01/rdf-schema, where the rdf terminology is defined.

Times support RDF, and it has achieved the status of a de facto standard for semantic representations on the web, together with the RDFS extension “http://www.w3.org/TR/rdf-schema/”, an RDF-based namespace introducing fundamental concepts such as classes. The obvious terminological proximity of RDF’s description (subject, predicate, object) and linguistics already suggests a good applicability to this field and to our task.

The prospective longevity of RDF as a de facto standard, and its widespread contemporary application, make RDF a suitable format for an I/O-interface of our data model. RDF helps disentangle the complex network of interoperating grammatical categories, intra- and interword relationships, influences from era, region or genre and provides an intuitive representation. We have to deal with utmost complexity in terms of variation, number of lexical entries and different relations between them in the prospective applications. Furthermore, RDF provides some means of consistency through validation. An RDF validation exists as a free web-interface provided by the W3C “http://www.w3.org/RDF/Validator/”. However, general logical problems like syllogisms or the liar paradox, which are a challenge even for the human mind, can not be unravelled by RDF. More general testing and human evaluation should therefore still be the irreplaceable last step of an overall evaluation process. A more fine grained ontological schema is OWL[3]. Unfortunately, the modelling is complex and the validation process is costly in terms of memory. An OWL extension of the RDF data model could be a next step for further development but does not guarantee the same degree of interoperability as RDF. Depending on the research goal, an OWL representation can, however, be chosen instead of RDF. Finally, the XML syntax of RDF ensures, that a certain nested structure is generated, helping to visually discriminate the entities in an RDF file at once.

Structure in RDF is generally modelled by the triple relations, but the RDFS extension provides additional possibilities, that need to be described. RDFS introduces among others some fundamental concepts: the concept of *class* (sub class) for subjects and objects and the concept of *sub property* for predicates. Typically, classes capture abstract concepts - for a lexicon, amongst others, the grammatical categories such as concrete cases. Sub class relations imply the presence of a hierarchical component structuring the abstract terminological entities’ network of relationships. Nominative, for instance is a case, a case is a grammatical category, hence nominative is in turn a grammatical category in our RDF representation. Concrete word forms or lemmas are assigned the most basic class attributes out of this network. This way, the hierarchy allows for inference then. If a concrete word form has the attribute “nominative” assigned by an rdf:type relation, it is also case marked, which the subClassOf relation for nominative implicitly invokes(nominative: subClassOf=”case”). The RDF XML specification is relatively straightforward and allows for bundles of triples of an appearance shown in the examples. The first example of “Sum Wan” being an author can be written in the above way or simply as: <rdf:Description rdf:about=“`http://example.org/rdf`” exampleorgnamespace:author=“Sum Wan”/>. At this point the terminological prerequisites for a description of our RDF representation have become evident. The next paragraph describes how the structural properties of the eLexicon described in section 2.1 are mapped onto the RDF format on a physical, i.e. the most concrete level.

For the eLexicon an abstraction of the lexical terminology has been undertaken. A namespace has

[3] `http://www.w3.org/TR/owl-semantics/`

been defined, based on RDF and RDFS, which relates resources, that is, grammatical categories. It is called elex-schema. An excerpt is shown in listing 2. In a second step, all concrete word forms and lemmas receive a distinct definition via bundled RDF triples as entries of the RDF lexicon file(s) (see listing 3), where memberships to categories defined in the elex-schema and the RDFS namespace describe the item grammatically and in terms of relations; the relation about which word form belongs to which lemma is exclusively encoded in the triple structure of lemmas internally (all triples with the predicate of elex:id_216 in listing 4). For each of the two files, genesis and peculiarities are described in the following section.

Listing 2: The Class Adjective.

```
<rdfs:Class rdf:ID="id_4">
 <rdfs:isDefinedBy rdf:resource="http://hudesktop.hucompute.org/elex-schema#"/>
 <rdfs:label>ADJ</rdfs:label>
 <rdfs:comment>adjective</rdfs:comment>
 <rdfs:subClassOf rdf:resource="#id_2"/>
</rdfs:Class>
```

Listing 3: Proximity to the RDFS schema definitions.

```
<rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-schema#Literal">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#"/>
  <rdfs:label>Literal</rdfs:label>
  <rdfs:comment>The class of literal values, e.g. textual strings and integers.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
```

The first step in transforming the abstract eLexicon model into an RDF XML representation was the definition of the terminological knowledge base, the grammatical relations in an elex-schema file. As can be seen from the examples 3 and 4, interoperability, longevity, understandability and usability have been augmented by ensuring similarity to the RDFS definitions themselves. Each entry is represented by a bundle of rdf triples. The bundle contains the RDFS elements isDefinedBy for the URI, label for a searchable, human readable label, comment and where applicable subClassOf, an anchor for the hierarchical position of the element. Most importantly, IDs are assigned to each entry in the schema file as well as in the entries file. This ensures that a relational data base representation and the RDF representation are easily convertible into each other and that elements are unique. As far as the hierarchy is concerned, the representation is quite transparent. The concepts class and subClassOf for grammatical categories used foremost as subjects stand aside the concepts of Property and subPropertyOf used foremost as predicates. These entail relations such as the lemma to word form relation and variation between lemmas or word forms. Properties restrict the class membership of the subjects and objects that are to be used with them by means of the RDFS attributes *range* and *domain*. A lemma to word form relation is a Property, where the domain/subject belongs to the class lemma and the object to the class word form. No other combination will be successfully validated (see listing 4). As RDF provides no means for logical negation, the representation does not aim at disallowing certain multiple assignments of attributes to one word form, which would not occurr together naturally. It does not - in other words - cluster the single grammatical properties that an inflectional language may combine into only one suffix, such as 3rd person, singular, present tense, but represents each feature in its own right. In this way, the model is also applicable to more than inflectional languages.

Listing 4: A relation. id_183 is word form, id_185 lemma.

```
<rdfs:Property rdf:ID="id_216">
 <rdfs:isDefinedBy rdf:resource="http://hudesktop.hucompute.org/elex-schema#"/>
 <rdfs:label>Lemma-to-word_form</rdfs:label>
 <rdfs:comment>Database-relation: Lemma-to-word form</rdfs:comment>
 <rdfs:subPropertyOf rdf:resource="#id_214"/>
 <rdfs:range rdf:resource="#id_183"/>
 <rdfs:domain rdf:resource="#id_185"/>
</rdfs:Property>
```

On a structural level it contents itself with the representation of direct hierarchical (directed) relations forming tree structures. The relations between word forms are represented by properties, and the same holds for the relations between lemmas.
Once the grammar has been defined, word forms can enter the lexicon. The second file is the representation of the lexicon. Word forms and lemmas are displayed in a different fashion:

Listing 5: An RDF example - a word form.

```
<rdf:Description rdf:about="int_id_7">
  <rdfs:label>abaestuare</rdfs:label>
  <rdf:type rdf:resource="&elex;id_183"/>
  <rdf:type rdf:resource="&elex;id_144"/>
  <rdf:type rdf:resource="&elex;id_113"/>
  <rdf:type rdf:resource="&elex;id_127"/>
 </rdf:Description>
```

Listing 6: An RDF example -a lemma.

```
<rdf:Description rdf:about="int_id_6">
        <rdf:type rdf:resource="&elex;id_185"/>
        <rdf:type rdf:resource="&elex;id_23"/>
        <rdfs:label>abaestuare@N</rdfs:label>
        <elex:id_216 rdf:resource="int_id_1"/>
        <elex:id_216 rdf:resource="int_id_2"/>
        <elex:id_216 rdf:resource="int_id_3"/>
        <elex:id_216 rdf:resource="int_id_4"/>
        <elex:id_216 rdf:resource="int_id_5"/>
    </rdf:Description>
```

The word forms are declared by reference to types defining their grammatical class memberships in the elex- schema file. The word abaestuare for example, which can mean “to wave down” gets the types word form (id_183), infinitive (id_144), active (id_113), present (id_127). Sequence and number of properties are not restricted. The grammatical reality of an entry (what is being marked) is only prescribed into the triple bundle inside the lexicon file. The model is therefore open to new categories entering, to new combinations and to changing linguistic paradigms. Also, unknown information on one of the attributes of a word form or lemma, which has been irretrievably lost for an ancient language’s many hapax legomena, can be left out. The lemmas are classified analogously but with different categories such as PoS etc. and flagged (@N for nouns, @V verbs etc.). Additionally lemma entries contain all related word forms by reference to their ids with the property id_216 which is the Lemma-to-word_form relation. While the categories assigned to word forms differ in terms of number and distribution according to the word class, the appearance of lemmas is more uniform in this representation.

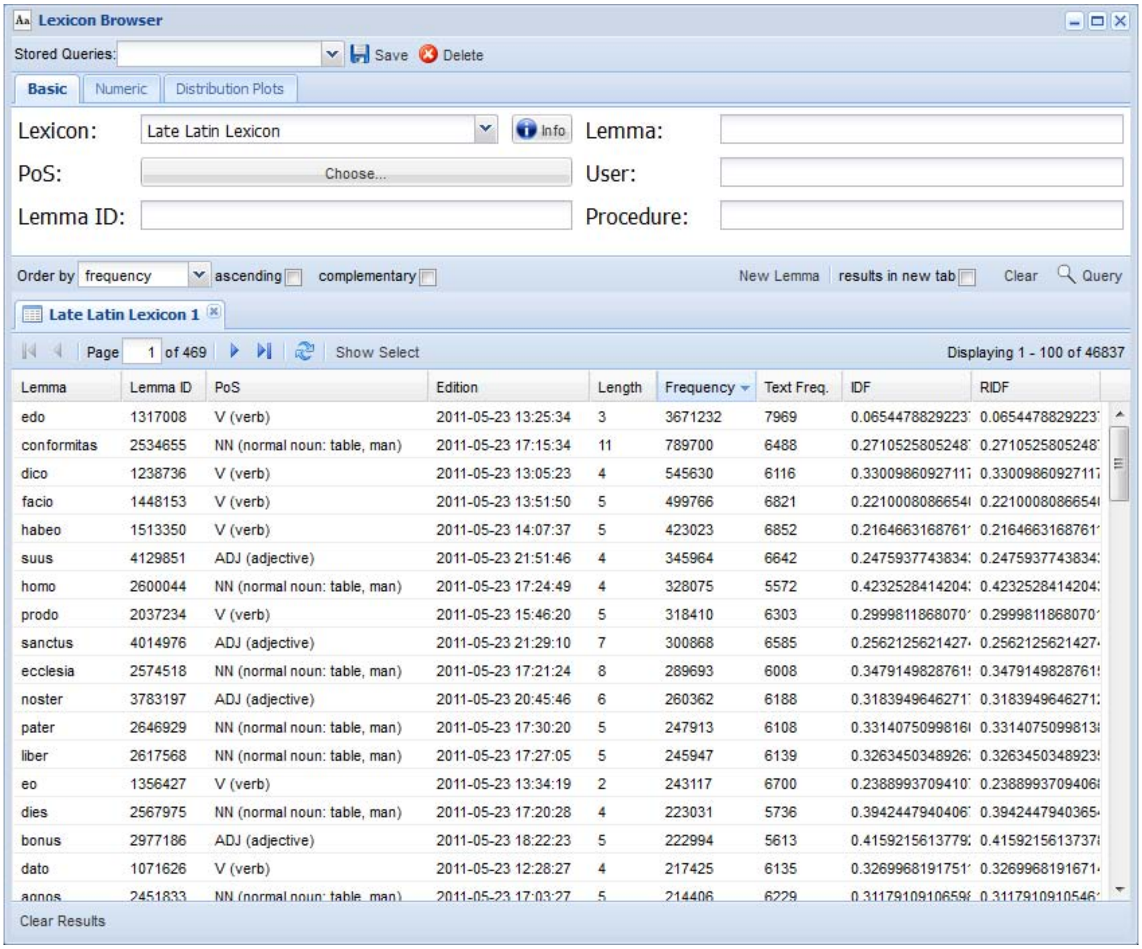
Summing up, the physical RDF XML representation of the eLexicon encompasses 2 major files, an elex-schema file wherein the grammatical categories and relation types are defined. The bundles of RDF triples constituting one entry are similar to the definitions in the rdfs schema definition itself. The second file contains lemmas and word forms (and can be split according to word class, number of entries etc. into multiple files). Word forms come with a definition of their grammatical features one by one relating to single definitions in the elex- schema file. Lemmas are equally attributed to grammatical categories, but fewer in number, additionally each lemma is connected to all its respective word forms by one triple for each such word form. Every grammatical category, word form and lemma possesses a unique id rendering the model convertible into a relational data base structure. The model is open to new entries on every level at any time providing a basic hierarchical structure without being too restrictive, e.g. one can leave out information that is not available for some word forms. Therefore it is especially applicable to ancient corpora.

2.4 eLexicon Browser

So far we have decribed the eLexicon from an informational perspective and described different approaches of modeling the lexicon data. In the following we focus on how an eLexicon instance can be browsed, searched and edited. It may seem trivial, but designing a lexicon interface to handle arbitrary languages and to keep the complexity of the underlying data model manageable is not an easy task. This section describes the eLexicon Browser, a web interface to lexicon instances which supports browsing, sophisticated searching as well as editing and plotting frequency distributions. The eLexicon Browser is build upon the eHumanities Desktop (Gleim and Mehler, 2010), a web-based Desktop Environment for humanists. It supports resource and access management on linguistic corpora, image collections and other research artefacts. On this basis application modules can be developed to process, browse and analyse resources. The eLexicon Browser is such a module.

A lexicon is managed like any other document within the eHumanities Desktop, which can be organized in repositories, shared with other users and exported as RDF XML. This allows research groups to explicitly define who can edit, read or manage a given lexical resource. Linguistic corpora can - if represented in TEI P5 - be used as a basis to compute some statistical data on lexicon entries like IDF, RIDF and (Text) Frequency. Lemma lists can be filtered in various ways. In addition to using wildcards and filtering by part of speech, desired ranges for the measure IDF, RIDF etc. can be defined. Figure 5 shows a screenshot of the lemma list of the Latin lexicon filtered. In this example the statistical data was computed based on the Patrologia Latina. The list is filtered to display only nouns, verbs and adjectives, and is sorted by frequency in descending order. The rank distribution of the lemmas with respect to a specific measure can also be plotted directly online.

Of course juggling with lemmas is only half the job. The interesting - and demanding - part is displaying and editing the morphology of a given lemma. Since the set of cases, persons, and tempora varies from language to language, the conjugation/declension tables cannot be hard-wired in terms of programming and database schemata. The contents and layout of the tables need to be rather flexible and extensible when new languages are being added. Since the eLexicon models morphological information of a word form as dimensions of a relation this can be solved quite easily. Plainly speaking, for each cell of a table the morphological information (i.e. set of dimensions) need



Stored Queries:

Lexicon: Late Latin Lexicon

PoS:

Lemma ID:

Lemma:

User:

Procedure:

Order by frequency ascending □ complementary □

results in new tab □

Late Latin Lexicon 1

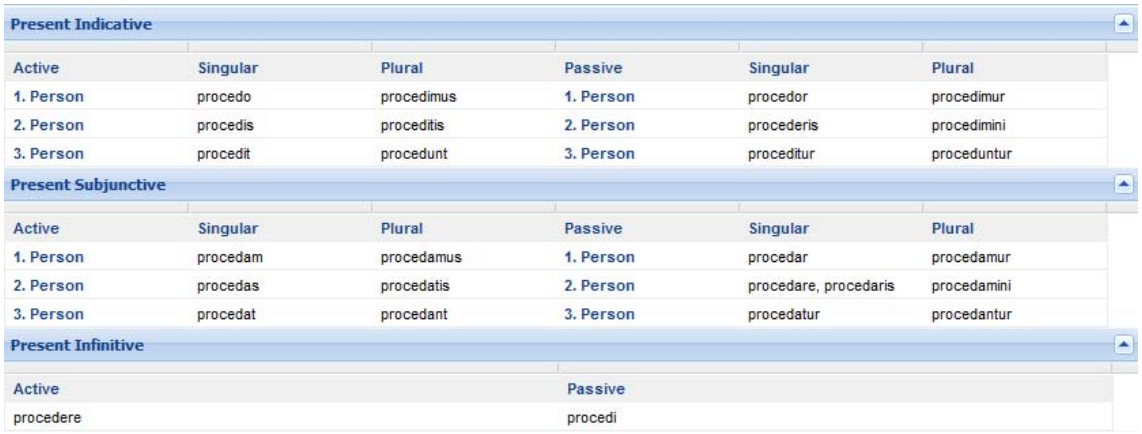
Page 1 of 469

Displaying 1 - 100 of 46837

Lemma	Lemma ID	PoS	Edition	Length	Frequency	Text Freq.	IDF	RIDF
edo	1317008	V (verb)	2011-05-23 13:25:34	3	3671232	7969	0.0654478829223	0.0654478829223
conformitas	2534655	NN (normal noun: table, man)	2011-05-23 17:15:34	11	789700	6488	0.2710525805248	0.2710525805248
dico	1238736	V (verb)	2011-05-23 13:05:23	4	545630	6116	0.3300986092711	0.3300986092711
facio	1448153	V (verb)	2011-05-23 13:51:50	5	499766	6821	0.2210008086654	0.2210008086654
habeo	1513350	V (verb)	2011-05-23 14:07:37	5	423023	6852	0.2164663168761	0.2164663168761
suus	4129851	ADJ (adjective)	2011-05-23 21:51:46	4	345964	6642	0.2475937743834	0.2475937743834
homo	2600044	NN (normal noun: table, man)	2011-05-23 17:24:49	4	328075	5572	0.4232528414204	0.4232528414204
prodo	2037234	V (verb)	2011-05-23 15:46:20	5	318410	6303	0.2999811868070	0.2999811868070
sanctus	4014976	ADJ (adjective)	2011-05-23 21:29:10	7	300868	6585	0.2562125621427	0.2562125621427
ecclesia	2574518	NN (normal noun: table, man)	2011-05-23 17:21:24	8	289693	6008	0.3479149828761	0.3479149828761
noster	3783197	ADJ (adjective)	2011-05-23 20:45:46	6	260362	6188	0.3183949646271	0.3183949646271
pater	2646929	NN (normal noun: table, man)	2011-05-23 17:30:20	5	247913	6108	0.3314075099816	0.3314075099813
liber	2617568	NN (normal noun: table, man)	2011-05-23 17:27:05	5	245947	6139	0.3263450348926	0.3263450348923
eo	1356427	V (verb)	2011-05-23 13:34:19	2	243117	6700	0.2388993709410	0.2388993709406
dies	2567975	NN (normal noun: table, man)	2011-05-23 17:20:28	4	223031	5736	0.3942447940406	0.3942447940365
bonus	2977186	ADJ (adjective)	2011-05-23 18:22:23	5	222994	5613	0.4159215613779	0.4159215613737
dato	1071626	V (verb)	2011-05-23 12:28:27	4	217425	6135	0.3269968191751	0.3269968191671
annos	2451833	NN (normal noun: table, man)	2011-05-23 17:03:27	5	214406	6229	0.3117910910659	0.3117910910546

Figure 5: Screenshot of eLexicon Browser with lemma list

to be specified. Dimensions (such as case) can be assigned to an entire table to avoid specifying it redundantly for each cell. The eLexicon Browser uses human-readable configuration files to declare the tables of a given language and what tables should be shown for a specific part of speech. Figure 6 shows an excerpt of the conjugation tables of the latin verb “procedo”.



Present Indicative

Active	Singular	Plural	Passive	Singular	Plural
1. Person	procedo	procedimus	1. Person	procedor	procedimur
2. Person	procedis	proceditis	2. Person	procederis	procedimini
3. Person	procedit	procedunt	3. Person	proceditur	proceduntur

Present Subjunctive

Active	Singular	Plural	Passive	Singular	Plural
1. Person	procedam	procedamus	1. Person	procedar	procedamur
2. Person	procedas	procedatis	2. Person	procedare, procedaris	procedamini
3. Person	procedat	procedant	3. Person	procedatur	procedantur

Present Infinitive

Active	Passive
procedere	procedi

Figure 6: Screenshot of the conjugation tables of procedo (excerpt)

If the user has the proper permissions he or she can also add and edit new entries directly using the conjugation/declension tables.[4]

3 eLexicon Instances

The eLexicon is actively used for various languages and corpora ranging from ancient languages to present day special lexicons. In the following we describe three instances: A Late Latin lexicon which has been composed by combining several sources, a lexicon of Avestan and a lexicon of the language that Hugo von Hofmannsthal[5] uses in his works.

3.1 Late Latin Lexicon

Although lexicographic work in the field of Late Latin can be traced back to the 17^{th} century with the *Glossarium mediæ et infimæ Latinitatis* of du Cange (1850), modern digital lexical resources for corpuslinguistic preprocessing are mostly limited to Classical Latin – that’s because the variability of Latin in the Middle Ages is quite a challenge in terms of data modelling.

An important focus of our work is the Patrologia Latina corpus (PL; Migne, 1865). With more than eight thousand Christian documents, compiled by Jacques Paul Migne from 1844 to 1855, it is one of the largest available corpora in Late Latin. The documents date from the 4^{th} to the early 13^{th} century and represent a wide variety of textual genres, including biographical texts, sermons, and contracts. Because of the large time range, the textual diversity and because the authors are widely dispersed

[4] However some of the editing functions are not completely implemented yet and the eLexicon browser is in ongoing development.

[5] Hugo von Hofmannsthal (1874-1929) was a multi-talented Austrian novelist and poet

geographically, the PL illustrates the development of Late Latin from the Late Antiquity to the high middle ages in a wide spectrum, even revealing its direction to the development of Early Romance[6].

This development, however, is rather subtle. As an established standard of Classical Latin is present in this period (the literary Latin of Cicero), documents of this era "are no longer written reflecting the way people spoke, but the way they were taught to write" (Clackson and Horrocks, 2007, p. 265), which makes it a language of written rather than spoken tradition (cf. Coseriu, 2008, p. 131f). The static, classical core of the language led to the image of Late Latin as a corpse with still-growing hairs and nails (Traube, 1911, p. 44). That means, although there is some development on many linguistic levels, like phonology, morphology, syntax, and vocabulary (Clackson and Horrocks, 2007, p. 272–284), the language does not develop as a whole and most of the novelties can be seen as non-systematic (Coseriu, 2008, p. 133). Imprints of these aspects of language change can be especially found in instances of so called *Vulgar Latin*, a term that refers to "the set of all the innovations and trends that turned up in the usage (...) of the Latin-speaking population who were little or not at all influenced by school education and by literary models" (Herman, 2000, p. 7). In the PL for example, the *Historia Francorum* by Gregory of Tours (6^{th} century) is seen to be such an instance.

To create a lexical resource to deal with this vocabulary, our first attempt (Mehler et al., 2011) was to merge existing digital dictionaries of Latin, namely from the *AGFL Grammatica Latina* project[7] (Koster, 2005), the *Perseus* project[8] (Smith et al., 2000), the *LEMLAT* project[9] (Passarotti, 2001), and *Words*[10]. These projects take rule-based approaches to morhological analysis, which means, they work best with systematic language patterns.[11] The most challenging tasks of combining these existing resources were dealing with different glossing rules, different sets of part of speech definitions, different rules of orthography (beginning with the treatment of the vowel "v" and the half vowel "u", where even the currently best known latin lexica differ – compare Glare et al. (1968) and Short and Lewis (1987)), and erroneous entries (introduced, for example, by the systematic generation of word forms that need to be treated as exceptional cases). The resulting eLexicon instance based on these resources (after cleaning and normalisation) is composed of 100,541 super lemmas, 116,286 lemmas and 2,836,327 word forms. However, this processing could not rule out all shortcomings of the sources, as the merging process grants only little control on the generation of the individual resource lexica. Therefore the Late Latin lexicon is under ongoing development.

The advanced digital preprocessing and analysis of historic corpora like the PL is attracting increasing interest in the scientific community. An example is the project IMPACT (Balk, 2010) which "aims to significantly improve access to historical text and to take away the barriers that stand in the way of the mass digitisation of the European cultural heritage." Another project is the one on historical semantics by Bernhard Jussen (Jussen et al., 2007) who explores the PL and related corpora as linguistic manifestations of socio-historic processes.

[6] Regarding the influence of Latin to the development of the Romance Languages, see Bonfante (1998).

[7] `http://www.agfl.cs.ru.nl/lat/`; last accessed: 7/13/2011.

[8] `http://www.perseus.tufts.edu/`; last accessed: 7/13/2011.

[9] `http://www.ilc.cnr.it/lemlat/`; last accessed: 7/13/2011.

[10] `http://users.erols.com/whitaker/words.htm`; last accessed: 7/13/2011.

[11] A similar approach on using multiple dictionaries as a resource is the "Database of Latin Dictionaries" by the Centre "Traditio Litterarum Occidentalium" (`http://brepolis.net/`), which even includes du Canges work. However, this database focuses on searching rather than corpuslinguistic preprocessing.

3.2 Avestan Lexicon

Avestan is an extinct Iranian sister language to Sanskrit. Religous texts of the old Iranian religion of Zoroaster are the only transmitted sources for it. The Avesta is a very small text corpus extremely complicated in the variants of its transmission (Cantera, 2010). Linguistically a lexicon for the Avesta mainly has to deal with the problem of uncertainty in classification of a word form. A lot of variants are not classifiable in an easy way and for quite some of the hapax legomena different scholars analyse them in different ways (Cantera, 1999). The eLexicon offers not only the possibility of avoiding aspects of a classification for a word form as mentioned above, applicable to word forms where the lack/loss of information on inflectional paradigms has for the time being rendered them impossible to analyse (to be supplemented once a classification has been undertaken). The eLexicon can also provide alternative scholarly classification information introducing a "scientific-classification-alternative" type. The entry is duplicated and, in addition to the classifications, is assigned the new alternative-type. At each cycle misclassifications can be eliminated. This makes it necessary for all the information to bundle tightly into closer relation packages, so that in case of a deletion these packages can be detected and removed easily. RDF also provides an excellent opportunity to bundle lexical information into packages. The RDF presentation might be used for deletions or insertions of bundles, once the project has reached the maintenance stage. At the moment, after having defined the eLex schema for Avestan, that is the relevant grammatical and lexical categories, a written lexicon is being digitized.

3.3 Hofmannsthal Lexicon

In cooperation with the Goethehaus Freies Deutsches Hochstift, the largest German critical edition for the oeuvre of an author shall be made accessible in an electronic way. The writer and poet, Hugo von Hofmannsthal, lived from 1874 to 1929 near Vienna, Austria. In order to make his literature scientifically analysable in a variety of ways encompassing an advanced search, the text corpus will be tagged and transformed into a TEI P5 format. Peculiarities of the authors style, the time period and the region he lived in prohibit the use of a conventional tagger for modern standard German and likewise the use of any modern standard German lexicon. These three sources of lexical variation lead in a multitude of cases to several different variants for one and the same word form. Furthermore, the required lexicon's complexity is high because poetic language is more diverse than other genres. Firstly, in the process of manifesting the contemporary rules for its written fixation, the standard German language underwent several stages of conventions and orthography reforms, the last of which went into effect in 2006, designed to unify a once chaotic jungle of conventions for the pairwise relations of Graphemes and Phonemes. As for the texts of the critical edition this reform as well as the last two reforms (1901 and 1876) affect the text corpus, because the volumes of the critical edition were released before and after the last reform, adhering to different standards; the other two reforms fall into the life and work period of the writer himself. A lexical expansion of all words into all their orthographic variants can use the rules that were fixed in the orthography conferences. For example a 'th', the conference decided, should be replaced by 't' in almost any case. An expansion process that aims at reproducing the older form, in order to map both variants onto the same lemma would reverse this rule and replace any 't' in a word with a 'th'. By such an expansion the lexicon can be filled with write- variants. However, once again, typically for short words rather than for long words it lies in the responsibility of the lexicon user to decide whether or not to input an expanded form. Any expanded form might coincide with an already present form of another lemma. This could trigger a wrong recognition. The eLexicon allows for homophones to enter as seperate entries, while a polysemy type can mark them as such. With the write-variant relation at hand, we can insert the thoroughly generated and filtered word forms into our eLexicon and duplicate their lexical properties from an existing entry if present. In this way the eLexicon deals with influences from time period and with changing orthographic paradigms (see example in figure 7). Secondly, each and every writer makes mistakes. Being exposed to a wide variety of orthographic conventions, facing an orthographic reform within his life and being a speaker and writer of a variety of interrelated European languages, Hugo von Hofmannsthal himself was no exception. The way to deal with 'mistakes' is similar to the orthographic variation. A formal grammar with expansion rules might be formulated, that, applied to each word, yields a number of different variants. In all these variants over generation might appear, that is variants so improbable, they might not even be produced by the most distracted writer. These forms however, as they are not likely to be matched, do not worsen the performance, but might occupy some additional memory. The eLexicon does not classify a variant explicitly as an error, because it is difficult to distinguish orthographic variants from errors, taking into account that there is some overlap. For instance, in the expression "the curse of time" the word "curse" could be chosen consciously or an error originally meaning "course" or even pointing to both meanings. The question arises, whether an error is an "error" in the sense of a wrong form, a slip of the pen, in opposition to a consciously chosen variant or a word play. Most certainly for writers, in literature and especially in poems, this consideration blurs the clarity of the concept of error; in the eLexicon, the word_form-write-variant-to-word_form relation is chosen for unique cases and lemma to lemma write variant for systematic cases; this leaves the interpretation of error vs. true variant to the scientific user. Thirdly, Hugo von Hofmannsthal was an Austrian citizen and his variety of German, was not free of Austrian German influences. This manifests itself again in lexical items that are not present in any modern German lexicon available. There are some resources for Austrian German however, which have been explored and integrated into the lexicon; additionally another formal grammar can produce Austrian German variants of German expressions in a regular way, especially when a morphological process such as forming a diminutive is involved. In the case of Austrian, the eLexicon allows for the input of word forms of different languages or sub-languages. In our case Austrian terms get a type "Austrian". They might be related however to the corresponding standard German expression through a relation of the respective lemmas to each other (see example in figure 7).

In this way, the Hugo von Hofmannsthal texts are another historical corpus that can be dealt with using the eLexicon structure.

4 Conclusion

We have discussed challenges of lexicon data models that have to deal with diachrony and variation. These problems do not only arise in long-term corpora such as the Patrologia Latina, but also in lexica which reflect, for example, the language use of a specific author. In this paper we proposed a multi layer approach to tackle the complexity of the requirements on the lexicon data model. We started

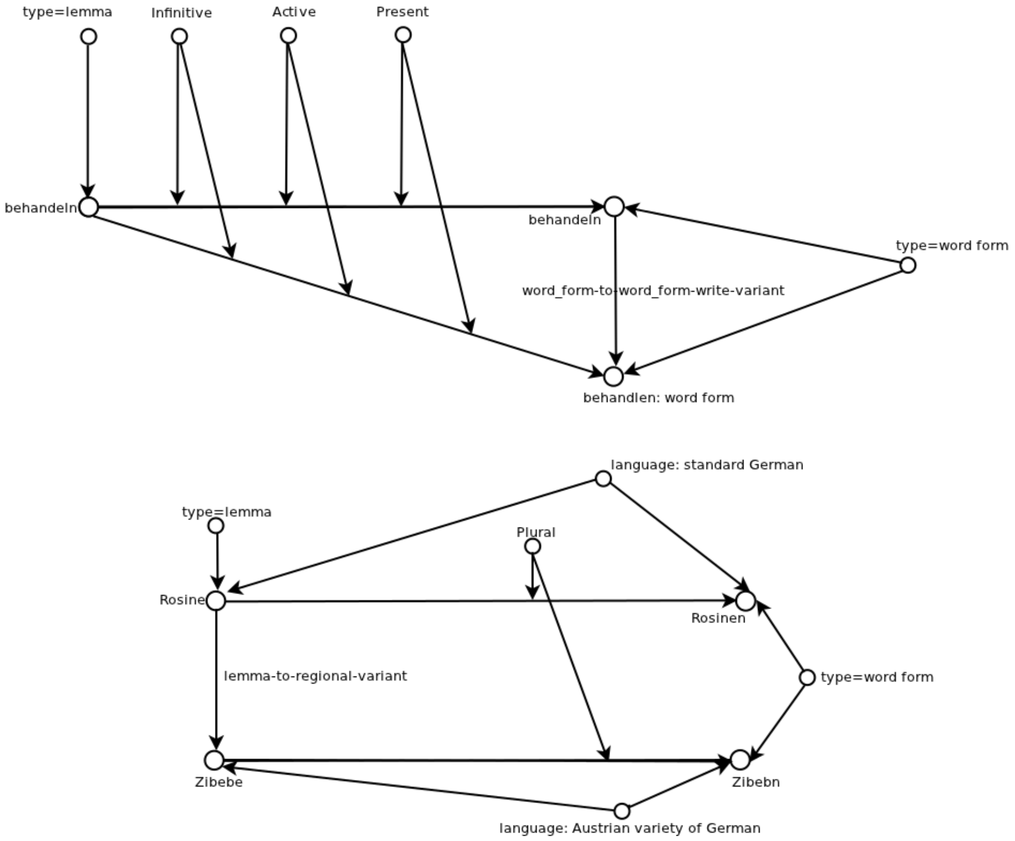


Figure 7: Example for “to treat” (writing variant) above and for “raisin” (variety variant) below.

with an abstract, conceptual perspective, the *eLexicon Data Model*. On this basis we described two logical data models which address different use cases but offer the same expressiveness. Section 2.2 introduced an extensible relational data model for lexica, whereas section 2.3 described an RDF based approach. Furthermore we have shown a web-based interface to browse and query lexicon data. Finally section 3 described three lexicon instances which are based on the eLexicon Data Model.

Current work is focussing on the improvement of the Late Latin lexicon instance and the extension of the user interface.

References

ANSI (1975). Interim report. fdt (bulletin of acm sigmod) 7:2. Technical report, ANSI/X3/SPARC Study Group on Data Base Management Systems.

Balk, H. (2010). Impact annual report 2009, version 1.0. `http://www.impact-project.eu`.

Bonfante, G. (1998). *The Origin of the Romance Languages. Stages in the Development of Latin*, volume 100 of *Bibliothek der klassischen Altertumswissenschaften: Reihe 2; N. F.* Universitätsverlag C. Winter, Heidelberg.

Cantera, A. (1999). Av. aγuriia- und siγuriia-: Zu einem umstrittenen lautgestzt im jungavestischen. *Münchener Studien zur Sprachwissenschaft*, 59:39–50.

Cantera, A. (2010). Rituales, manuscritos y ediciones del avesta: Hacia una nueva edición de los textos avésticos de la liturgia larga. *Boletín de la Sociedad Española de Iranología*, 1:28–42.

Clackson, J. and Horrocks, G. (2007). *The Blackwell History of the Latin Language*. Blackwell Publishing Ltd, 2 edition.

Coseriu, E. (2008). *Lateinisch – Romanisch. Vorlesungen und Abhandlungen zum sogenannten Vulgärlatein und zur Entstehung der romanischen Sprachen*, volume 1 of *Schriften des Eugenio-Coseriu-Archivs*. Gunther Narr Verlag, Tübingen.

du Cange, C. d. F. (1840–1850). *Glossarium mediæ et infimæ Latinitatis conditum a Carolo Dufresne domino du Cange auctum a monachis O. S. B. cum supplementis integris D. P. Carpenterii et additamentis Adelungii et aliorum digessit G. A. L. Henschel*, volume 1–7. Firmin Didot fratres, Paris, France. First print 1678.

Glare, P., Souter, A., and Wyllie, J., editors (1968). *Oxford Latin Dictionary*. Clarendon Press, Oxford, UK.

Gleim, R. and Mehler, A. (2010). Computational linguistics for mere mortals – powerful but easy-to-use linguistic processing for scientists in the humanities. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, Valletta/Malta.

Herman, J. (2000). *Vulgar Latin*. The Pennsylvania State University Press, University Park, Pennsylvania. Translated by Roger Wright; First published in 1967 as Le latin vulgaire.

Hitzler, P., Krötzsch, M., Rudolph, S., and Sure, Y. (2008). *Semantic Web*. Springer.

Jussen, B., Mehler, A., and Ernst, A. (2007). A corpus management system for historical semantics. *Sprache und Datenverarbeitung. International Journal for Language Data Processing*, 31(1-2):81–89.

Kenekayoro, P. T. (2011). Semantic web - the future of the web. *African Journal of Mathematics and Computer Science Research*, 4(3):113–116.

Koster, C. H. A. (2005). Constructing a parser for latin. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing CICLing*, number 3406 in Lecture Notes in Computer Science, pages 48–59. Springer, Mexico City, Mexico.

Mehler, A., Diewald, N., Waltinger, U., Gleim, R., Esch, D., Job, B., Küchelmann, T., Pustylnikov, O., and Blanchard, P. (2011). Evolution of Romance language in written communication: Network analysis of late Latin and early Romance corpora. *Leonardo*, 44(3).

Mehler, A., Gleim, R., Ernst, A., and Waltinger, U. (2008). WikiDB: Building interoperable wiki-based knowledge resources for semantic databases. *Sprache und Datenverarbeitung. International Journal for Language Data Processing*, 32(1):47–70.

Migne, J. P. (1844–1865). *Patrologiae cursus completus: Series latina*, volume 1–221. Chadwyck-Healey, Cambridge.

Passarotti, M. (2000/2001). Development and perspectives of the latin morphological analyser LEMLAT. *Linguistica computazionale*, XX/XXI:397–414.

Short, C. and Lewis, C. T., editors (1987). *A Latin Dictionary*. Clarendon Press, Oxford, UK.

Smith, D. A., Rydberg-Cox, J. A., and Crane, G. R. (2000). The perseus project: A digital library for the humanities. *Literary and Linguistic Computing*, 15(1):15–25.

TEI Consortium, editor (2010). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium, http://www.tei-c.org/Guidelines/P5/.

Traube, L. (1911). *Einleitung in die Lateinische Philologie des Mittelalters*. Verlag C. H. Beck, München, reprint 1965 edition.

W3C (2010). Resource description framework (rdf). `http://www.w3.org/RDF/`. Last access: May 25, 2011.