Noname manuscript No.
(will be inserted by the editor)

Assessing and Improving the Quality of SKOS Vocabularies

Osma Suominen · Christian Mader

Received: date / Accepted: date

Abstract Controlled vocabularies are increasingly made available on the Web of Data using the SKOS ontology. Assessment of vocabulary quality is important for determining the suitability of vocabularies for reuse in applications and for improving vocabulary development processes. We define 26 *quality issues*, i.e., computable functions that expose potential quality problems. In an analysis of a representative set of 24 SKOS vocabularies, we found all of them to contain structural errors and/or other quality problems. We propose a set of *correction heuristics* which we have used to automatically correct a significant proportion of the identified problems. Our reference implementations of these methods, the quality assessment tool qSKOS and the quality improvement tool Skosify, are available for reuse as open source software.

Keywords Controlled Vocabularies · Linked Data · Semantic Web · Quality Assessment · Data Quality

1 Introduction

Controlled vocabularies such as taxonomies, classifications, subject headings and thesauri [4] are widely used in search and retrieval settings to, e.g., improve search results or provide assistance to the user in the exploration of knowledge bases [15]. In recent years, many organizations have published their controlled vocabularies online using the Simple Knowledge Organization System (SKOS) ontology [28]. As an example, library classifications have been published as SKOS vocabularies, allowing various library catalogs to be published as Linked Data and then integrated using RDF tools [10, 27, 40], enabling applications such as semantic information retrieval over multiple datasets [11].

However, published Linked Data is often plagued by quality issues such as modeling errors and inconsistent, malformed or missing data [13, 18, 19]. Hundreds of published SKOS vocabularies can be found online [3], and many of them contain defects that hinder their effective use [2] and their applicability for various types of applications [31].

Quality assessment of SKOS vocabularies is important for several reasons. First, vocabulary developers can now reuse some of the many available SKOS vocabularies and integrate them with their own vocabularies. However, they need to assess the quality of a candidate vocabulary to decide whether to adopt it. Second, development of a controlled vocabulary is often a long-running, error-prone process. Many contributors work on the vocabulary consecutively or collaboratively, possibly introducing errors such as redundant concepts or conflicting relations among concepts [15]. If quality issues are assessed at all, the checks performed are often tailor-made for a specific data format or development tool (e.g., [32, 12]), lacking compatibility with other approaches.

We address these issues by the definition of a framework for automated assessment and correction of common potential quality issues in SKOS vocabularies. Our contributions encompass:

O. Suominen
Semantic Computing Research Group
Aalto University, Department of Media Technology
E-mail: osma.suominen@aalto.fi
tel. +358504316155

C. Mader
Multimedia Information Systems Group
University of Vienna, Faculty of Computer Science
E-mail: christian.mader@univie.ac.at

- Definition of 26 automatically computable quality checking functions that are based on existing work in the field of controlled vocabulary development and Linked Data publication. They identify elements in the vocabulary that possibly cause a degradation of quality (*quality issues*).
- Methods to automatically correct 12 of these quality issues.
- Study of 24 vocabularies available in SKOS format to find out about occurrences of quality issues and the effectiveness of automatic correction.
- Freely available, open source reference implementations of the quality assessment and improvement tools.

We address the following research questions:

1. How can the quality of SKOS vocabularies be automatically measured?
2. To what extent are existing SKOS vocabularies on the Web affected by quality problems?
3. Can the quality of SKOS vocabularies be improved using an automated process?

The research reported in this article is a continuation of our earlier research [26,41]. Compared to our previous studies, we present a more comprehensive list of quality checking functions, employ a more systematic selection of vocabularies and use three different tools to analyze and process them: the *qSKOS* quality analysis toolkit[1], the *Skosify* vocabulary processor[2], and the *PoolParty online SKOS Consistency Checker*[3] (hereafter known as the *PoolParty checker*). In addition, we use *qSKOS* to measure the effectiveness of the automated quality issue correction heuristics implemented by *Skosify*.

The remainder of this article is structured as follows: In Section 2 we provide an overview of existing data quality assessment approaches, especially related to SKOS vocabularies. We present our method for defining quality issues, the *qSKOS* and *Skosify* tools we have developed, our test data set and our evaluation setup in Section 3. In Section 4, we formulate a set of 26 quality issues for assessing SKOS vocabularies. We then evaluate 24 SKOS vocabularies of various domains and sizes using three tools, and present the results in Section 5. In Section 6, we attempt to automatically correct a subset of the identified problems in the vocabularies using the *Skosify* tool and present the results of reevaluating the corrected vocabularies. We then discuss the relevance and validity of our findings in Section 7 and conclude our article with suggestions for future work in Section 8.

[1] `https://github.com/cmader/qSKOS/`

[2] `http://code.google.com/p/skosify/`

[3] `http://demo.semantic-web.at:8080/SkosServices/check`

2 Background and Related Work

2.1 Data Quality in General

The problem of *vocabulary quality* is closely related to *data quality*, and has been discussed in data and information systems research [7]. Pipino et al. argue that dealing with data quality should involve both "subjective perceptions of the individuals" and "objective measurements based on the data" [35]. We see our work as a contribution to the latter.

2.2 Controlled Vocabulary Quality

Typical application areas of controlled vocabularies are classification, indexing, auto-completion, query reformulation, or glossary functionality. These areas impose specific requirements on vocabulary features, such as structure, availability, and documentation [31]. Quality aspects of controlled vocabularies have already been discussed in standardized guidelines [1,33], manuals [4, 15,17,6], tutorials [39], and scholarly articles [12,24]. Quality assessment in these most often relies on manual, precise analysis of individual statements in the data, as in Soergel's tutorial [39]. Our work builds on this literature, but focuses on the less intellectually loaded checks, which can be automated to assist vocabulary users or publishers.

Kless and Milton [24] provide an overview of intrinsic abstract measurement constructs for thesaurus evaluation that are presumably useful as thesaurus quality measures. They are classified into five areas, namely *Concept-related*, *Term-related*, *Structure-related*, *Documentation*, and *Overall*. In our work we utilize a similar classification but, in contrast to the work of Kless, persue a more formal approach in defining quality issues. Some constructs given by Kless are designed for intellectual evalulation (e.g., "Conceptual clarity" or "Complexity"), whereas some can serve as starting point for defining formalized measurements (e.g., "Documentation completeness" or "Structural correctness"). Thus, our contributions in this article refine some of these measurements in a formal way.

2.3 Quality of SKOS Vocabularies

An early guide for creating SKOS vocabularies by Miles et al. [29] already stressed the importance of error checking and validation, but the validation is only performed on the RDF syntax level. Van Assem's description of a method for converting existing thesauri to SKOS [6] mentions the difficulty of SKOS validation, which has

since been addressed by later revisions of the SKOS specification and the development of validation tools.

The SKOS specification does not mention the notion of quality, but lists in total six integrity conditions [28], each of which is a statement that defines under which circumstances data are consistent with the SKOS model. For example, "a resource has no more than one value of `skos:prefLabel` per language tag". Tools that can check whether these conditions are met are available. Two of the six conditions are defined formally in the OWL representation of SKOS and can therefore be validated by OWL reasoners such as OWLIM[4]. The *PoolParty checker* performs many checks on SKOS vocabularies, including the SKOS integrity conditions. It was originally developed to determine if the vocabulary can be imported into the *PoolParty thesaurus editor* [38]. The W3C used to host a similar online SKOS validation service, but it was not kept up to date with the evolution of SKOS, and is no longer available. However, implementations vary, particularly in the level of support for RDFS and OWL reasoning, SKOS inference rules, and the extent to which they implement the informally specified SKOS integrity conditions. Thus, the results of these checks cannot always be directly compared.

Abdul Manaf et al. [3] have surveyed the landscape of SKOS vocabularies available on the Web and analyzed their high level structural properties, such as the number of hierarchy levels and in- and outgoing links to other concepts. However, they give no statement about how each of these measurements affect the quality of a vocabulary. The same authors have also identified three types of common problems (*slips*) in SKOS vocabularies as well as possible ways to correct them (*patches*) [2]. They can be found by OWL reasoning and are partly based on the axioms defined in the SKOS reference ontology. However, the number of proposed slips and corresponding patches is quite small and mostly concerned with making the SKOS vocabularies processable using an OWL reasoner, not with the quality of the intellectual content of the vocabulary.

The authors of the SKOS version of the STW Thesaurus of Economics describe the use of SPARQL queries to find inconsistencies in SKOS vocabularies [32]. However, they do not describe the checks they used in detail.

2.4 Quality of Linked Data Sets

More general validation services for RDF and Linked Data have also been developed. The *W3C RDF Validation Service*[5] can be used to verify the syntax of RDF documents. The *Vapour* [8] and *RDF:Alerts* [18] systems are online validation tools intended to spot problems in Linked Data. For OWL datasets, the *Pellet ICV* reasoner re-interprets OWL axioms with integrity constraint semantics. SPARQL Inferencing Notation[6] (SPIN) is a SPARQL-based language which can be used to specify integrity constraints for RDF data [14]. The *TopBraid Composer*[7] suite is one tool supporting SPIN-based validation, and it includes a SPIN ruleset that implements testing of the SKOS integrity conditions.

A recent and thorough survey of general RDF and Linked Data validation tools is given by Hogan et al. [18] identifying four categories of common errors and shortcomings in RDF documents. Also, Heath et al. [16] summarize best practices for publishing data on the Web. The *Pedantic Web Group*[8] is an online community of practitioners who help to correct errors in the publication of RDF data. However, to our knowledge, none of these tools and approaches have any specific support for SKOS vocabularies.

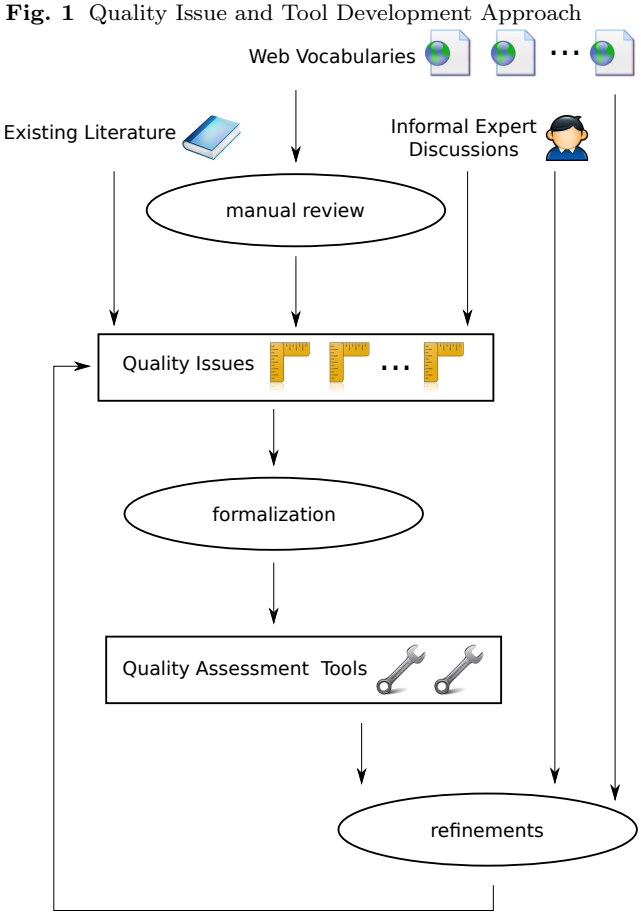
2.5 Ontology Evaluation, Repair, and Improvement

Ontology evaluation, i.e., measuring the quality of an ontology, has been discussed extensively by Vrandecic [44]. However, the author focuses on RDF datasets and ontologies in general. While some of these criteria, such as consistent tagging of literals, are relevant for SKOS vocabularies, these need to be completed by considering SKOS-specific properties.

Repairing problematic constructs in OWL ontologies has been extensively discussed by Kalyanpur [23]. Ovchinnikova et al. propose a method for solving inconsistencies in ontology design by rewriting problematic axioms [34]. Horridge et al. present methods for explaining inconsistencies in OWL ontologies [21]. The OOPS! pitfall scanner is an OWL ontology evaluation tool that provides the user with guidelines about how to solve the issues it has found [37]. However, these OWL-related methods are only partially relevant to SKOS vocabularies, because not all of the SKOS integrity conditions and other quality measures can be expressed using OWL axioms[9]. To our knowledge, automatic correction methods intended specifically for SKOS vocabulary constructs have not been proposed earlier, except in our own earlier work [41].

[4] `http://www.ontotext.com/owlim`
[5] `http://www.w3.org/RDF/Validator/`
[6] `http://spinrdf.org`
[7] `http://topquadrant.com/products/TB_Composer.html`
[8] `http://pedantic-web.org`
[9] In particular, neither OWL nor OWL 2 include any means to express the integrity condition S14: *"A resource has no more than one value of `skos:prefLabel` per language tag."*

Fig. 1 Quality Issue and Tool Development Approach



3 Materials and Methods

3.1 Quality Issues and Tools

The process for building the tools to address the first research question (*How can the quality of SKOS vocabularies be automatically measured?*) is depicted in Figure 1. First, it is essential to define what the notion of *quality* means for SKOS, i.e. what distinguishes a "good" vocabulary from a "bad" one. To accomplish this, we utilized three main sources for our research, namely (i) review of existing literature on design, construction and evaluation of controlled vocabularies, (ii) informal discussion with experts in the field of vocabulary development and publication, and (iii) manual review of currently published vocabularies on the Web. This allowed us to extract 26 *quality issues*, i.e. computable quality functions that identify resources and relations which *possibly* cause quality problems. We formalized these issues and implemented tools that take a SKOS vocabulary file as input and output a report on the identified quality issues. The underlying strategy for most issues (except for *Missing In-links* and *Broken Links*) in our analysis process is to treat a vocabulary as a self-contained entity, resembling the closed world assumption, as is generally done when validating RDF data.

3.1.1 Defining Quality Issues

We identified an initial set of possible quality issues in SKOS vocabularies by focusing on issues that can be measured automatically. Some graph measures, such as hierarchy depth or node centrality, have been omitted due to lack of evidence on their general influence on vocabulary quality. We published our findings in the *qSKOS* wiki[10] and requested feedback from experts via public mailing lists[11], a workshop publication [25] and informal face to face discussions. Based on the received responses, we translated a subset of these issues into computable quality checking functions. Each function takes a given SKOS vocabulary and an optional vocabulary namespace as input and finds all resources that match the corresponding quality issue. We included all the quality issues assessed by the *PoolParty checker* into our list of quality issues.

We divided the issues into three categories: *Labeling and Documentation Issues*, *Structural Issues*, and *Linked Data Specific Issues*. We did not assign grades of severity to the issues, because such a judgement is highly dependent on the context and intended application of the vocabulary. However, five of the quality issues correspond to the SKOS integrity conditions and violations of these could be considered more severe than the other quality issues. The final list of 26 quality issues is presented in detail in Section 4.

3.1.2 Developing Assessment Tools

Some features of the tools we use in this study, *qSKOS*, *Skosify*, and the *PoolParty checker*, are summarized in Table 1. The tools we have developed, *qSKOS* and *Skosify*, follow different approaches:

qSKOS [25,26] has been designed to apply and (re-)interpret vocabulary quality recommendations to the requirements of Web vocabularies, aiming to contribute a general catalog of quality issues (cf. Section 3.1.1) that is usable for various domains and use-cases. *qSKOS* aims to provide both short and detailed reports on vocabulary resources and relations that cause potential quality problems which can be addressed by human experts. Secondly, by providing an API, it is designed to be integrated

[10] `https://github.com/cmader/qskos/wiki`
[11] e.g., `public-esw-thes@w3.org` and `public-lod@w3.org`

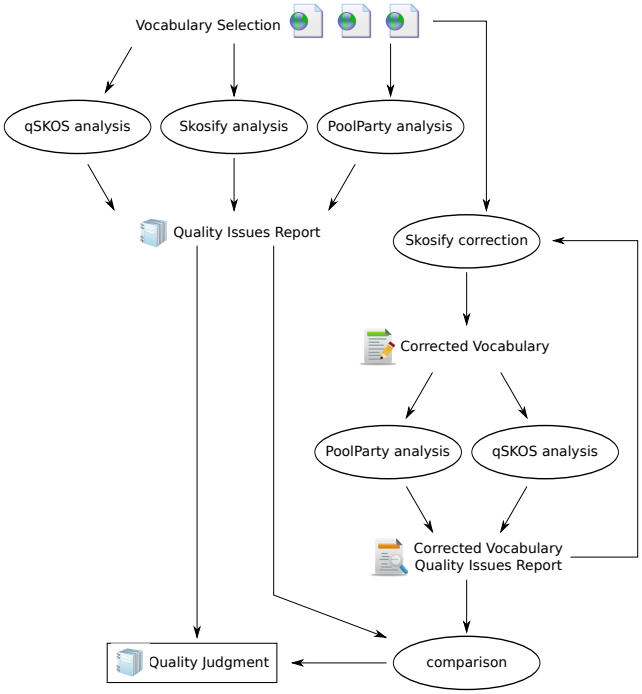
Table 1 Details on the tools used for vocabulary analysis.

	qSKOS	Skosify	PoolParty checker
Version	0.9.5	0.6	(current online version)
License	GPLv3	MIT	Proprietary
Implementation Language	Java	Python	(unknown)
Inference Support	SKOS specific rules, RDFS	SKOS specific rules, RDFS	SKOS, RDFS, OWL
Web Interface	no	yes	yes
API	yes (Java)	yes (online REST)	no
Home Page URL	`https://github.com/cmader/qSKOS/`	`http://code.google.com/p/skosify/`	`http://demo.semantic-web.at:8080/SkosServices/check`

into existing vocabulary development environments to provide continual feedback on the current quality state of the vocabulary. *qSKOS* has been developed and published as open source software.

Skosify [41] originates from an effort to automatically convert vocabularies expressed as RDFS and OWL into well-structured SKOS vocabularies. The focus of the tool lies in *automatic correction* of quality issues that have commonly been observed by reviewing vocabularies on the Web. These include checking for adherence to the SKOS integrity conditions and correcting problems whenever possible. *Skosify* also considers structural issues or labeling shortcomings that can be automatically repaired, such as cycles in hierarchical relations or superflous whitespace in labels. Many vocabularies published on the ONKI ontology service [43] are automatically processed with *Skosify* as a part of the publication process. A public web interface[12] is available which can be used to validate and process small (up to 20MB) vocabularies. *Skosify* has also been published as open source software.

Fig. 2 Quality Assessment and Improvement Approach



3.2 Quality Assessment and Improvement

The process for building the tools to address the second (*To what extent are existing SKOS vocabularies on the Web affected by quality problems?*) and third (*Can the quality of SKOS vocabularies be improved using an automated process?*) research questions is depicted in Figure 2. To find out if and how the identified quality issues are relevant in a practical setting, we compiled a set of SKOS vocabularies from different domains and of different size and access policy. The selection process is explained in detail in Section 3.2.1. From this set, we analyzed each vocabulary with the *PoolParty checker*, *Skosify* and *qSKOS* tools. The resulting *Quality Issues Report* provided a basis for further manual analysis of possible causes and implications of the potential quality problems (indicated as *Quality Judgment* box in the figure). In addition to the vocabulary analysis, we performed automatic correction of each vocabulary using *Skosify*. To get an understanding about the usefulness and effectiveness of the tool, we analysed the corrected vocabulary with the *PoolParty checker* and *qSKOS* in the next step. Based on this analysis we manually adjusted the correction settings for *Skosify* to achieve the best possible correction results. Afterwards, we compared the quality issue reports of the corrected and the uncorrected vocabularies and incorporated it in our *Quality Judgment* (cf. Section 7).

3.2.1 Vocabulary Set

To collect a suitable data set of SKOS vocabularies, we used the following procedure. First, in order to ensure a wide coverage of domains, we looked for vocabularies in each of the seven categories of the Linked Open

[12] `http://demo.seco.tkk.fi/skosify/`

Table 2 Vocabularies selected for further analysis. The Concepts column shows the number of authoritative SKOS concepts in the vocabulary, i.e., concepts whose URI is within the URI namespace of the vocabulary.

Abbrev	Vocabulary Name	Version	Domain	Size	Concepts
ODT	Open Data Thesaurus	2012-09-11	Cross-domain	small	107
Eurovoc	The EU's multilingual thesaurus	4.3	Cross-domain	medium	6797
UMBEL	UMBEL Vocabulary and Reference Concept Ontology	1.05	Cross-domain	large	26389
GeoNames	GeoNames Ontology	3.01	Geographic	small	680
NYTL	New York Times Locations	2012-09-11	Geographic	(medium)	1920
EARTh	The Environmental Applications Reference Thesaurus	2012-08-30	Geographic	large	14351
Reegle	Clean Energy and Climate Change Thesaurus	2012-09-28	Government	small	1447
IPSV	Integrated Public Sector Vocabulary	2.00	Government	medium	4732
LVAk	Austrian Armed Forces Thesaurus	0.9	Government	large	13411
PXV	Peroxisome Knowledge Base	1.6	Life sciences	small	1686
GEMET	The GEneral Multilingual Environmental Thesaurus	2012-09-11	Life sciences	medium	5209
SNOMED	SNOMED clinical terms (French)	3.5-VF-20091001	Life sciences	large	102614
IPTC	IPTC NewsCodes / Media Topic	2012-09-12	Media	small	2061
NYTP	New York Times People	2012-09-10	Media	medium	4979
GTAA	Gemeenschappelijke Thesaurus Audiovisuele Archieven	2010-08-25	Media	large	171991
UNESCO	UNESCO nomenclature for fields of science and technology	2012-12-20	Publications	small	2509
STW	STW Thesaurus for Economics	8.10	Publications	medium	6789
LCSH	Library of Congress Subject Headings	2012-03-01	Publications	large	408923
AGROVOC	United Nations Agricultural Thesaurus	2012-07-26	Publications	large	32291
RAMEAU	French National Library subject headings	2009-04-23	Publications	large	207272
DDC	Dewey Decimal Classification	2012-09-28	Publications	large	251977
SSW	Social Semantic Web Thesaurus	2012-09-11	User-generated content	small	1943
Plant	Plant Building Vocabulary	2012-09-11	User-generated content	medium	3246
DBpedia	DBpedia Categories	3.8	User-generated content	large	865902

Data cloud domain classification[13]. For each domain, we then selected one small (up to 3000 concepts), one medium-size (3001 to 10000 concepts) and one large (more than 10000 concepts) SKOS vocabulary. This two-dimensional matrix gave us 21 slots to fill with a vocabulary. For each slot, we used three data sources to select a prominent, recently updated (not older than 2009) SKOS vocabulary that was available for download or SPARQL access from (i) the Datasets page[14] of the SKOS wiki, which mentions approximately 40 sources, some of which contain several SKOS vocabularies; (ii) the SKOS vocabularies listed in the Data Hub data catalog[15], approximately 150 datasets tagged `format-skos` or `skos`; and (iii) the survey of SKOS vocabularies by Abdul Manaf et al. [3], containing 478 vocabularies. We also included vocabularies that are not available for public access, e.g., the *LVAk thesaurus* used by the Austrian army and the *Peroxisome Knowledge Base*[16] that was provided to us as a RDF dump.

This procedure gave us 20 SKOS vocabularies, with the slot for a medium size vocabulary in the Geographical domain still unfilled as we couldn't find a suitable vocabulary using those criteria. We chose to use the *New York Times Locations* vocabulary instead, which has 1920 concepts and is thus relatively large, although not large enough for the medium-size category. Finally, we chose to include all the very large vocabularies, having more than 100000 concepts, regardless of their domain: *DBpedia Categories*, the *Dewey Decimal Classification*, *GTAA*, *LCSH*, *RAMEAU* and *SNOMED CT*. The final set of 24 vocabularies is shown in Table 2.

We downloaded each vocabulary that was provided as one or more RDF files and also included any mappings provided by the vocabulary publisher. For vocabularies that were only available as SPARQL endpoints, we used a script[17] to query for all the triples in the store and serialized them into files. We converted each vocabulary to a single merged file in Turtle syntax using the `rdfcat` utility from the *Apache Jena*[18] distribution. Some vocabularies were further pre-processed[19] before they could be successfully analyzed.

Detailed statistics about each vocabulary are summarized in Table 3 and discussed in Section 5.2.1.

3.2.2 Analysis of Vocabularies

To gain an understanding of the current quality of SKOS vocabularies published online, we analyzed the 24 vocabularies described in Section 3.2.1 using the *PoolParty*

[13] `http://wifo5-03.informatik.uni-mannheim.de/lodcloud/state/#domains`

[14] `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`

[15] `http://datahub.io/`

[16] `http://www.peroxisomekb.nl/`

[17] The script, `sparqldump.py`, is included in the *Skosify* distribution.

[18] `http://jena.apache.org`

[19] Missing namespace declarations were added manually for UMBEL. In NYTL, the invalid language tag `fr_1793` was manually changed into `fr-1793` in order to comply with BCP47 and the Turtle specification. In Reegle, an unparseable line in the original RDF dump was manually removed. For GEMET, the source file containing Arabic labels was excluded as it contained labels with improper Unicode encoding that caused the Jena toolkit to fail in parsing it.

Table 3 Vocabulary statistics as determined by *qSKOS*.

	All Concepts	Authoritative Concepts	Concept Labels	Semantic Relations	Concept Schemes	Collections	HTTP URIs
ODT	233	107	326	688	6	0	493
Eurovoc	6797	6797	457788	18491	128	0	403936
UMBEL	26393	26389	88621	72338	0	0	26922
GeoNames	680	680	3241	0	9	0	179
NYTL	1920	1920	1920	0	1	0	64462
EARTh	26137	14351	30403	48094	1	0	26161
Reegle	2952	1447	3665	32553	12	0	5480
IPSV	4732	4732	7945	13843	3	0	4772
LVAk	13411	13411	17250	16346	0	0	13414
PXV	2112	1686	3628	2695	1	0	2770
GEMET	14112	5209	165890	22129	1	79	14198
SNOMED	102614	102614	150964	265712	1	0	10
IPTC	2061	2061	1128	2241	0	0	2066
NYTP	4979	4979	4979	0	1	0	29342
GTAA	171991	171991	178776	50892	9	0	172006
UNESCO	2509	2509	7512	5740	1	0	2516
STW	25107	6789	58441	145433	3	0	25171
LCSH	503476	408923	750219	659885	1	0	503538
AGROVOC	52893	32291	624776	86647	1	0	666574
RAMEAU	355158	207272	470392	465751	0	0	1648701
DDC	251977	251977	158162	302331	70	0	284819
SSW	2656	1943	3487	17171	10	0	4389
Plant	6492	3246	3581	28618	3	0	11405
DBpedia	865902	865902	862826	1730458	0	0	865905

checker, the *qSKOS* quality analysis toolkit and the *Skosify* tool to find possible quality issues.

We used the *PoolParty checker* to analyze those vocabularies that could be expressed in a single RDF file that was below the 20MB size limit of the PoolParty checker. This ruled out the largest vocabularies: Eurovoc, GTAA, LCSH, AGROVOC, RAMEAU, DDC, and DBpedia. UMBEL and SNOMED were further condensed[20] before validation in order to stay below the 20MB size limit of the *PoolParty checker*.

We also used the *qSKOS* tool to analyze all the 24 vocabularies, looking for possible quality issues. On the largest vocabularies, the *Missing In-links* and *Broken Links* were performed on randomly sampled subsets of the concepts for performance reasons. The reported values were extrapolated from the measurements on the subset and are marked with an asterisk in Table 8. For ODT and STW, an URI pattern was explicitly specified to identify authorative concepts.

The value for *Extra Whitespace in Labels* was determined from the output of the *Skosify* processing described below, as the measure is not implemented in *qSKOS*.

[20] The Turtle files were condensed by removing extra whitespace, including all indentation, and using short 0–2 character namespace prefixes.

3.2.3 Correcting Problems in Vocabularies

To find out whether some of the identified problems could be automatically corrected, we developed a set of *correction heuristics* to address a subset of the quality issues. We chose to attempt to correct twelve issues where a straightforward algorithmic correction was determined to be feasible and the goal of the correction was clear. This ruled out, e.g., corrections involving the addition of labels, documentation properties, or relationships between concepts, because it would be difficult for a computer to choose the correct additions to make. We also concentrated on frequently occurring quality issues that affected many different vocabularies.

These correction heuristics are similar in spirit to the *patches* described by Abdul Manaf et al. [3], though our heuristics operate on the level of SKOS vocabulary constructs instead of correcting more general OWL modeling issues. We implemented these heuristics, described in detail in Section 6.1, in the *Skosify* tool (cf. Table 4). The heuristics were presented in our earlier work [41], but the implementation has since been refined to better address issues detected by *qSKOS*.

Some of the corrections are optional or require some parameters. We chose suitable correction settings for each vocabulary. The selection process and the chosen settings are described in Section 6.2.

After applying the correction heuristics to each vocabulary, we evaluated the effect of the heuristics by reanalyzing the corrected vocabularies using the *PoolParty checker* and *qSKOS* tools. The results of the evaluation are described in Section 6.3.

4 Quality Issues

The quality issues we have defined are summarized in Table 4. In the following, we explain the origins and design rationale for each quality issue and explain how the corresponding quality checking function works. For better readability and due to lack of space we provide only semi-formal definitions and refer to the source code of the *qSKOS* tool for further details.

4.1 Definitions

For the purpose of this work, we define a SKOS vocabulary as follows:

Definition (SKOS Vocabulary) *Let a SKOS vocabulary be a tuple of the form* $V = \langle IR, C, AC, SR, LV, CS \rangle$, *with*
$IR = I_{CEXT}(\texttt{rdfs:Resource}^{\mathcal{I}})$ *being the set of* ***resources****,*

Table 4 Our quality issues related to the SKOS integrity conditions and the issues detected by the *PoolParty checker*, and support for the quality issue in our *qSKOS* and *Skosify* tools. When the same or very similar quality issue has been discussed in the SKOS reference or in our own earlier work, this has been indicated by references to the respective publications.

Categ.	Criterion name [earlier work]	SKOS	PoolParty checker	qSKOS	Skosify
Labeling and Documentation Issues	Omitted or Invalid Language Tags [26,41]	-	Missing Language Tags	assessed	corrected
	Incomplete Language Coverage [26]	-	-	assessed	-
	Undocumented Concepts [26]	-	-	assessed	-
	Overlapping Labels [26]	-	-	assessed	-
	Missing Labels [41]	-	Missing Labels	-	partially corrected
	Inconsistent Preferred Labels [28,41]	S14	Consistent Use of Labels	assessed	corrected
	Disjoint Labels Violation [28,41]	S13	Consistent Use of Labels	assessed	corrected
	Extra Whitespace in Labels [41]	-	-	-	corrected
Structural Issues	Orphan Concepts [26]	-	-	assessed	-
	Disconnected Concept Clusters [26]	-	-	assessed	-
	Cyclic Hierarchical Relations [26,41]	-	-	assessed	corrected
	Valueless Associative Relations [26]	-	-	assessed	-
	Solely Transitively Related Concepts [26]	-	-	assessed	corrected
	Omitted Top Concepts [26]	-		assessed	partially corrected
	Unmarked Top Concepts [41]	-	Loose Concepts		corrected
	Top Concepts Having Broader Concepts [26]	-	-	assessed	-
	Unidirectionally Related Concepts	-	-	assessed	corrected
	Relation Clashes [28,41]	S27	Consistent Usage of Semantic Relations	assessed	corrected
	Mapping Clashes [28,41]	S46	Consistent Usage of Mapping Properties	assessed	-
	Disjoint Classes Violation [28,41]	S9,S37	Disjoint OWL Classes	-	partially corrected
Linked Data Specific Issues	Missing In-links [26]	-	-	assessed	-
	Missing Out-links [26]	-	-	assessed	-
	Broken Links [26]	-	-	assessed	-
	Undefined SKOS Resources [26]	-	-	assessed	-
	HTTP URI Scheme Violation	-	-	assessed	-
	Invalid URIs [41]	-	Valid URIs	-	-

$C \subseteq IR$ *with* $C = I_{CEXT}(\texttt{skos:Concept}^{\mathcal{I}})$ *being the set of* ***concepts****,*
$AC \subseteq C$ *being the set of* ***authoritative concepts****, which are all concepts that are identified by URIs in the vocabulary namespace, as opposed to concepts from other vocabularies that have been referenced in the RDF graph,*
$SR = I_{EXT}(\texttt{skos:semanticRelation}^{\mathcal{I}})$ *being the set of* ***semantic relations*** *associating concepts with one another,*
$LV \subseteq I_{CEXT}(\texttt{rdfs:Literal}^{\mathcal{I}})$ *being the set of untyped* ***plain literals****, and*
$CS = I_{CEXT}(\texttt{skos:ConceptScheme}^{\mathcal{I}})$ *being the set of* ***concept schemes****.*

Further, we let V be the fully entailed RDFS interpretation of the underlying RDF graph. We enrich V by entailment of `owl:inverseOf` *properties as well as instances of* `owl:TransitiveProperty` *and* `owl:SymmetricProperty` *defined by the formal OWL semantics of SKOS [28].*

4.2 Labeling and Documentation Issues

4.2.1 Omitted or Invalid Language Tags

SKOS defines a set of properties that link resources with RDF literals, which are plain text natural language strings with an optional language tag. This includes the labeling properties `rdfs:label`, `prefLabel`[21], `altLabel`, `hiddenLabel` and also SKOS documentation properties, such as `note` and subproperties thereof. Literals should be tagged consistently [44], because omitting language tags or using non-standardized, private language tags in a SKOS vocabulary could unintentionally limit the result set of language-dependent queries. A SKOS vocabulary can be checked for omitted and invalid language tags by iterating over all resources in IR and finding those that have labeling or documentation property relations to plain literals in LV with missing or invalid language tags, i.e., tags that do not comply with the syntactic rules of BCP47[22] and language codes not listed in the ISO 639[23] standard.

4.2.2 Incomplete Language Coverage

The set of language tags used by the literal values linked with a concept should be the same for all concepts. If this is not the case, appropriate actions like, e.g., splitting concepts or introducing scope notes should be taken by the creators. This is particularly important for applications that rely on internationalization and translation use cases. Affected concepts can be identified

[21] Typographical note: words set in typewriter style that don't include a namespace prefix, such as `Concept` and `prefLabel`, refer to terms defined by SKOS [28].
[22] `http://tools.ietf.org/html/bcp47`
[23] `http://www.iso.org/iso/language_codes`

by first extracting the global set of language tags used in a vocabulary from all literal values in LV, which are attached to a concept in C. In a second iteration over all concepts, those having a set of language tags that is not equal to the global language tag set are returned.

4.2.3 Undocumented Concepts

Svenonius [42] advocates the "inclusion of as much definition material as possible" and the SKOS Reference [28] defines a set of "documentation properties" intended to hold this kind of information. To identify all undocumented concepts, we iterate over all authoritative concepts in AC and collect those that do not use any of these documentation properties.

4.2.4 Overlapping Labels

The SKOS Primer [22] recommends that "no two concepts have the same preferred lexical label in a given language when they belong to the same concept scheme". This issue could affect application scenarios such as autocompletion, which proposes labels based on user input. Although these issues are acceptable for some thesauri, we generalize the above recommendation and search for all concept pairs with their respective `prefLabel`, `altLabel` or `hiddenLabel` property values meeting a certain similarity threshold defined by a function $sim : LV \times LV \rightarrow [0, 1]$. The default, built-in similarity function checks for case-insensitive string equality with a threshold equal to 1. Overlapping labels can be found in a two-staged algorithm. First, for each literal value lv in LV, *qSKOS* finds the set of all labeled concepts $LC \subseteq C$ that are related to lv by either `prefLabel`, `altLabel`, or `hiddenLabel` properties. In the second stage, we return those lv that have an associated set of LC with a cardinality > 1 as overlapping labels.

4.2.5 Missing Labels

The *PoolParty checker* finds (i) all concepts in C that don't have `prefLabel` relations, and (ii) all concept schemes in CS which don't have `rdfs:label` relations to literals in LV. This check is not performed by *qSKOS*.

4.2.6 Inconsistent Preferred Labels

The integrity condition S14 in the SKOS reference documentation states that "A resource has no more than one value of `skos:prefLabel` per language tag, *and no more than one value of* `skos:prefLabel` *without [a] language tag.*". The latter part of this definition is only present in the comments of `prefLabel` in the SKOS RDF Schema[24]. For every resource ir in IR, *qSKOS* finds sets of literal values $PL \subseteq LV$ that are related to ir by the `prefLabel` property. In a second iteration, every pair of $pl \times pl$ with pl in PL and belonging to the same ir is checked for identical language tags. Each of these pairs constitutes a violation of the integrity condition S14 and is thus returned by *qSKOS*, alongside with the affected resource ir.

4.2.7 Disjoint Labels Violation

Integrity condition S13 is defined as "`skos:prefLabel`, `skos:altLabel` and `skos:hiddenLabel` are pairwise disjoint properties." Similar to the *Overlapping Labels* issue described above, for each literal value lv in LV, *qSKOS* finds the set of all labeled resources $LR \subseteq IR$ that are related to lv by at least two of the properties `prefLabel`, `altLabel`, or `hiddenLabel`. Every labeled resource in lr in a set LR with cardinality > 1 is then returned as a violation of the integrity condition S13.

4.2.8 Extra Whitespace in Labels

Extra (i.e., leading and trailing) whitespace is unlikely to carry meaning and may cause problems when the vocabulary is, e.g., stored in a database or used for information retrieval, particularly when exact string matching is performed. Such extra whitespace is likely an artifact of conversion from another textual format such as XML or CSV, or it may originate in the text fields of graphical user interfaces used for vocabulary editing, where whitespace is typically invisible. We check for whitespaces in the literal values of all resources in IR that are related by the SKOS label properties `prefLabel`, `altLabel` and `hiddenLabel` or SKOS documentation properties including `note` and all its sub-properties.

4.3 Structural Issues

4.3.1 Orphan Concepts

This issue is motivated by the notion of "orphan terms" in the literature [17], i.e., terms without any associative or hierarchical relationships. Checking for such terms is common in thesaurus development and also suggested by the ANSI/NISO Z39.19 guidelines [33]. Since SKOS is concept-centric, we define an orphan concept as being a concept that has no semantic relation $sr \in SR$ with any other concept. Although it might have attached lexical labels, it lacks valuable context information, which can be essential for retrieval tasks such as query expansion.

[24] `http://www.w3.org/2009/08/skos-reference/skos.rdf`

Orphan concepts in a SKOS vocabulary can be found by iterating over all elements in C and selecting those without any semantic relation to another concept in C.

4.3.2 Disconnected Concept Clusters

A vocabulary can be split into separate clusters because of incomplete data acquisition, deprecated terms, accidental deletion of relations, etc. This can affect operations that rely on navigating a connected vocabulary structure, such as query expansion or suggestion of related terms. Disconnected concept clusters are identified by first creating an undirected graph that includes all non-orphan concepts (as defined above) as nodes and all semantic relations SR as edges. Tarjan's algorithm [20] can then be applied to find all connected components, i.e., all sets of concepts that are connected together by (chains of) semantic relations.

4.3.3 Cyclic Hierarchical Relations

This issue is motivated by Soergel et al. [39] who suggest a "check for hierarchy cycles" since they "throw the program [into] a loop in the generation of a complete hierarchical structure". Also Hedden [17], Harpring [15] and Aitchison et al. [4] argue that there exist common hierarchy types such as "generic-specific", "instance-of" or "whole-part" where cycles would be considered a logical contradiction. Cyclic relations can be found by constructing a graph with the set of nodes being C and the set of edges being all `broader` relations.

4.3.4 Valueless Associative Relations

The ISO/DIS 25964-1 standard [1] suggests that terms that share a common broader term should not be related associatively if this relation is only justified by the fact that they are siblings. This is advocated by Hedden [17] and Aitchison et al. [4] who point out "the risk that thesaurus compilers may overload the thesaurus with valueless relationships", having a negative effect on precision. This issue can be checked by identifying concept pairs $C \times C$ that share the same broader or narrower concept while also being associatively related by the property `related`.

4.3.5 Solely Transitively Related Concepts

Two concepts that are explicitly related by `broaderTransitive` and/or `narrowerTransitive` can be regarded a quality issue because, according to the SKOS Reference [28], these properties are "not used to make assertions". Transitive hierarchical relations in SKOS are meant to be inferred by the vocabulary consumer, which is reflected in the SKOS ontology by, for instance, `broader` being a subproperty of `broaderTransitive`. This issue can be detected by finding all concept pairs $C \times C$ that are directly related by `broaderTransitive` and/or `narrowerTransitive` relationships but not by (chains of) `broader` and `narrower` subproperties.

4.3.6 Omitted Top Concepts

The SKOS model provides concept schemes, which are a facility for grouping related concepts. This helps to provide "efficient access" [22] and simplifies orientation in the vocabulary. In order to provide entry points to such a group of concepts, one or more concepts can be marked as top concepts. Concept schemes with omitted top concepts can be detected by iterating over all concept schemes in CS and collecting those that do not occur in relations established by the properties `hasTopConcept` or `topConceptOf`.

4.3.7 Unmarked Top Concepts

This issue is closely related to *Omitted Top Concepts*. Unmarked top concepts are concepts that are not marked as top concepts (i.e., by having incoming `hasTopConcept` or outgoing `topConceptOf` relationships) in any `ConceptScheme`, and have no `broader` relationships pointing to other concepts. This issue is checked by the *PoolParty checker*, where it is called "Loose Concepts". It is not detected by *qSKOS*.

4.3.8 Top Concept Having Broader Concepts

Allemang et al. [5] propose to "not indicate any concepts internal to the tree as top concepts", which means that top concepts should not have broader concepts. Affected resources are found by collecting all top concepts that are related to a resource via a `broader` statement and not via `broadMatch`—mappings are not part of a vocabulary's "intrinsic" definition and a top concept in one vocabulary may perfectly have a broader concept in another vocabulary.

4.3.9 Unidirectionally Related Concepts

Inclusion of the complete set of reciprocal and symmetric relations can increase recall of queries in systems where no inferencing is or can be used. On the other side, explicit assertion of inferable facts can be seen as redundant. We define a tuple $V' = \langle IR, C, AC, SR', LV, CS \rangle$ in the same way as in definition 4.1, but with the constraint that $SR' \subseteq SR$ does not contain the mentioned

OWL entailments, i.e., we do not enrich the underlying RDF graph with inferable relations. *qSKOS* finds all pairs of resources in $IR \times IR$ that are related by SKOS property relations with specified inverse or symmetric relations but do not explicitly assert these relations.

4.3.10 Relation Clashes

The SKOS integrity condition S27 states that the associative relationship "`skos:related` is disjoint with the property `skos:broaderTransitive`". Two concepts that are in the same hierarchical transitive closure (as inferred by `broaderTransitive` or `narrowerTransitive` relations) must not be associatively related by the `related` property. To find pairs of "clashing" resources, *qSKOS* in a first step creates a directed hierarchy graph, containing all resources in IR that are related by one of the skos hierarchical properties (`broader`, `broaderTransitive`, `broadMatch` and their inverse counterparts). In a second step, all pairs of associatively (`related`, `relatedMatch`) connected concepts are selected. A relation clash is reported if there exists a path in the hierarchy graph between these pairs of concepts.

4.3.11 Mapping Clashes

The SKOS integrity condition S46 states that the mapping relationship "`skos:exactMatch` is disjoint with each of the properties `skos:broadMatch` and `skos:relatedMatch`." Accordingly, *qSKOS* reports all pairs of concepts that are related by both the `exactMatch` property and one of the `broadMatch`, `narrowMatch`, or `relatedMatch` properties.

4.3.12 Disjoint Classes Violation

The SKOS integrity conditions specifying class disjointness axioms, S9 ("`skos:ConceptScheme` is disjoint with `skos:Concept`") and S37 ("`skos:Collection` is disjoint with each of `skos:Concept` and `skos:ConceptScheme`"), are checked by the *PoolParty checker*, but not by the current version of *qSKOS*.

4.4 Linked Data Specific Issues

4.4.1 Missing In-links

When vocabularies are published on the Web, SKOS concepts become linkable resources. Estimating the number of in-links can indicate the importance of a concept. Many concepts without in-links may indicate a quality problem. We estimate the number of in-links by iterating over all elements in AC and querying the Sindice[25] and DataHub[26] SPARQL endpoints for triples containing the URI of the concept in the object part. Empty query results are indicators for missing in-links.

4.4.2 Missing Out-links

SKOS concepts should also be linked with other related concepts on the Web, "enabling seamless connections between data sets" [16]. This issue identifies the set of all authoritative concepts that have no links to other resources on the Web. It can be computed by iterating over all elements in AC and returning those that are not linked with any non-authoritative resource. Unlike *Missing In-links*, utilization of dataset registries is not necessary because out-links can be identified locally by comparing URI namespaces.

4.4.3 Broken Links

As discussed by Popitsch and Haslhofer [36], broken links are RDF resources that return HTTP error responses or no response at all when being dereferenced. An erroneous HTTP response in that case can be defined as a response code other than 200 after possible redirections. Just as in the "document" Web, these broken links hinder navigability also in the Web of Data and should therefore be avoided. Broken links are detected by iterating over all resources in IR, dereferencing their HTTP URIs, following possible redirects, and including unavailable resources in the result set.

4.4.4 Invalid URIs

This issue is closely related to the one discussed above. It targets resources with syntactically invalid URIs, i.e., URIs containing invalid characters such as whitespace. We list this issue separately because it is addressed by the *PoolParty checker* and only partly by *qSKOS*. Syntax checking for URIs is not performed by *qSKOS*. However, it can identify most invalid URIs by the lookup performed when checking for *Broken Links*.

4.4.5 Undefined SKOS Resources

The SKOS model is defined within the namespace `http://www.w3.org/2004/02/skos/core#`. However,

[25] `http://sindice.com/` indexes the Web of Data, which is composed of pages with semantic markup in RDF, RDFa, Microformats, or Microdata. Currently it covers approximately 230M documents with over 11 billion triples.

[26] `http://datahub.io/` is a "community-run catalogue" of currently 5045 datasets, many of them following the Linked Data guidelines.

some vocabularies use resources from within this namespace, which are unresolvable for two main reasons: vocabulary creators minted new terms within the SKOS namespace instead of introducing them in a separate namespace, or they use deprecated SKOS elements such as `subject`. Undefined SKOS resources can be identified by iterating over all resources in IR and returning those that (i) are contained in the list of deprecated resources[27] or (ii) are identified by a URI in the SKOS namespace but are not defined in the current version of the SKOS ontology.

4.4.6 HTTP URI Scheme Violation

The second principle of Tim Berners-Lee's article on Linked Data[28] encourages the use of (dereferencable) HTTP URIs as names for things described in the dataset. This way datasets can be interlinked, making possible, e.g., queries spanning multiple datasets. *qSKOS* finds and returns the set of all URIs at the subject part of the vocabulary's RDF triples that have a schema part other than `http` or `https`.

5 Analysis of Existing Vocabularies

In this section, we present the results of analyzing our test vocabularies using the *PoolParty checker*, the *qSKOS* tool and, for the *Extra Whitespace in Labels* check, also *Skosify*.

5.1 PoolParty Checker Validation Results

The results of validating the 17 vocabularies using the *PoolParty checker* are summarized in Table 5.

According to the *PoolParty checker*, all of the vocabularies used valid URIs. Six vocabularies had problems with missing language tags. Five vocabularies were missing human-readable labels for concepts or other resources of the vocabulary. Eight vocabularies contained loose concepts. These problems were classified as warnings in the *PoolParty checker* reports.

The last four checks of the *PoolParty checker* represent mandatory checks which are based on the integrity conditions in the SKOS specification. There were no detected problems involving OWL class disjointness axioms. Twelve vocabularies had problems involving the consistent use of labels, an example of which is illustrated in Figure 3(a). Reegle was the only vocabulary which had problems in the consistency of mapping properties. Semantic relations were inconsistent in nine of the vocabularies, examples of which are illustrated in Figures 3(b) and 3(c). Of the 17 vocabularies analyzed, four (IPTC, NYTP, UNESCO and Plant) passed all of the mandatory checks. The remaining 13 vocabularies, 76% of all analyzed vocabularies, were inconsistent with the SKOS integrity conditions according to the *PoolParty checker*.

5.2 qSKOS Quality Analysis Results

In this section we concentrate on giving examples that illustrate typical or curious findings. For further information the reports of the analysis results can be downloaded from the *Skosify* project site[29], which also includes both original and corrected versions of the non-restricted vocabularies.

5.2.1 Vocabulary Statistics

Table 3 summarizes some basic statistical properties of our vocabulary selection, such as the number of concepts and authoritative concepts, concept labels (i.e., `prefLabel`, `altLabel`, and `hiddenLabel` relations involving concepts), semantic relations (i.e., pairs of resources related by a subproperty of `semanticRelation`), and URIs that use the HTTP scheme.

From these properties we can see that approximately 3,000 DBpedia Categories concepts are missing labels (e.g., `Category:South_Korean_social_scientists`), which is a consequence of missing natural language descriptions in some Wikipedia categories. Also, many concepts in DDC are not labeled in natural language but have a `notation` literal defined instead.

We can also determine the type of the vocabulary from the number of `semanticRelation`s to some extent. GeoNames, NYTL and NYTP are mainly intended as authoritative lists and don't define, e.g., hierarchical or associative relations between concepts.

The reason for only ten found HTTP URIs in SNOMED is that the concepts are identified by URI fragments (e.g., `http://Snomed3_5.fr#C-7087`) which are to be evaluated on the client side and thus treated as one URI (`http://Snomed3_5.fr`) by *qSKOS*.

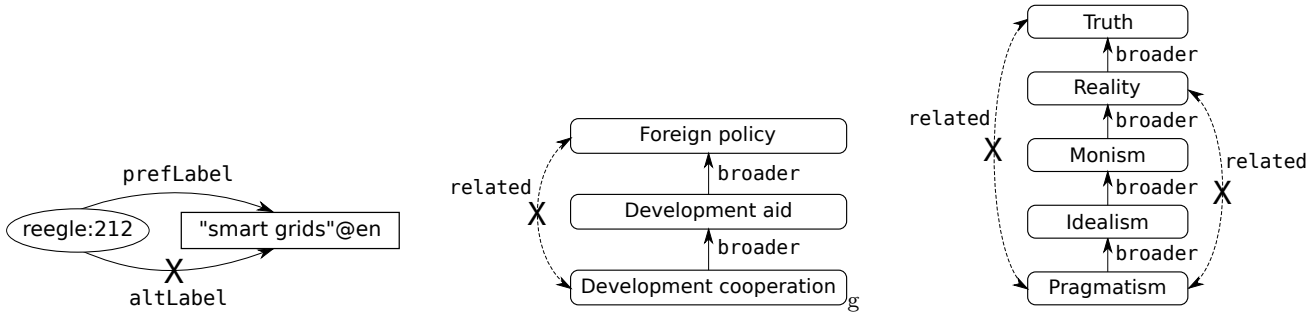
5.2.2 Labeling and Documentation Issues

Table 6 shows the result of our vocabulary analysis focusing on labeling and documentation related issues using *qSKOS* and *Skosify*. We found this kind of issues in all reviewed vocabularies.

[27] See `http://www.w3.org/TR/skos-reference/#namespace`
[28] `http://www.w3.org/DesignIssues/LinkedData.html`
[29] `http://code.google.com/p/skosify/downloads/list`

Table 5 Validation and Correction Results when using the *PoolParty checker*. The last four columns represent mandatory checks (corresponding to the SKOS integrity conditions) that must be passed for the vocabulary to be considered valid by the *PoolParty checker*. When an arrow symbol (→) is shown, the values before and after the arrow represent, respectively, the analysis result for the original vocabulary and the vocabulary after processing it with the *Skosify* tool. When no arrow is shown, the analysis result was unchanged. The *Skosify* corrections are discussed in more detail in Section 6.

	Valid URIs	**Missing Language Tags**	**Missing Labels**	**Loose Concepts**	**Disjoint OWL Classes (S9, S37)**	**Consistent Use of Labels (S13, S14)**	**Consistent Use of Mapping Properties (S46)**	**Consistent Use of Semantic Relations (S27)**
ODT	pass	pass	pass	pass	pass	**fail**→pass	pass	pass
UMBEL	pass	25794→pass	pass→**fail**	pass	pass	**fail**→pass	pass	pass
GeoNames	pass	pass	pass→**fail**	pass	pass	**fail**→pass	pass	pass
NYTL	pass	pass	pass	1920→pass	pass	**fail**	pass	pass
EARTh	pass	pass	**fail**	2687→pass	pass	**fail**→pass	pass	**fail**→pass
Reegle	pass	pass	pass	2→pass	pass	**fail**	**fail**	**fail**→pass
IPSV	pass	pass	**fail**	pass	pass	**fail**→pass	pass	**fail**→pass
LVAk	pass	13411→pass	pass	69→pass	pass	pass	pass	**fail**→pass
PXV	pass	1684→pass	**fail**	7→pass	pass	**fail**→pass	pass	**fail**→pass
GEMET	pass	3→pass	pass	109→pass	pass	**fail**→pass	pass	**fail**→pass
SNOMED	pass	102599→pass	**fail**	pass	pass	**fail**→pass	pass	**fail**→pass
IPTC	pass	pass	pass	pass	pass	pass	pass	pass
NYTP	pass	pass	pass	4979→pass	pass	pass	pass	pass
UNESCO	pass	pass	pass	pass	pass	pass	pass	pass
STW	pass	2→pass	**fail**	pass	pass	**fail**→pass	pass	**fail**→pass
SSW	pass	pass	pass	9→pass	pass	**fail**→pass	pass	**fail**→pass
Plant	pass	pass	pass	pass	pass	pass	pass	pass



(a) Overlapping labels in Reegle

(b) Related concepts within the same hierarchy in the STW Thesaurus

(c) Related concepts within the same hierarchy in LCSH

Fig. 3 Examples of overlapping labels and disjoint semantic relations. Crosses (X) mark relationships that were eliminated by *Skosify*. Figure adapted from earlier work [41].

Omitted or Invalid Language Tags can be observed in 14 of the 24 vocabularies. In ODT this issue only occurs in three blank nodes of the VoID dataset descriptor describing `void:TechnicalFeature`s. This is also the case for Plant, Reegle, and SSW which all were created with the *PoolParty Thesaurus Manager*.

Eurovoc describes 218 countries which have an `altLabel` consisting of two characters (e.g., "PT" for the Portuguese Republic) without a language tag. Additionally, one language tag is missing for the preferred label of the `ConceptScheme` definition.

PXV and LVAk omit language tags with their labeling properties, LCSH with documentation properties (e.g., `note`, `editorialNote`, `example`). STW uses many `@x-other` language tags, which are considered invalid by *qSKOS*, and additionally does not use language tags with two instances of `definition`, which have apparently been copied from the SKOS RDF schema.

SNOMED completely omits language tags for concepts. They are only used for the description and license statement of the vocabulary, expressed with the `dc:description` and `dc:rights` properties.

Table 6 Validation and correction results using the *qSKOS* quality analysis toolkit, part 1: *Labeling and Documentation Issues*. The figure for *Extra Whitespace in Labels* was determined using the *Skosify* tool.

	Omitted or Invalid Language Tags	Incomplete Language Coverage	Undocumented Concepts	Overlapping Labels	Inconsistent Preferred Labels	Disjoint Labels Violation	Extra Whitespace in Labels
ODT	3→0	16	35	2	0	1→0	0
Eurovoc	219	6370	5341	62	0	0	2
UMBEL	25793→0	0	2848	5207→5226	2→0	1→0	522
GeoNames	0	43	60	162	1→0	0	0
NYTL	0	0	1862	0	0	0	0
EARTh	10→0	313	7840	2100→2103	0	69→0	310
Reegle	3→0	1450	3	22	0	3→0	52
IPSV	0	0	4551	0	0	21→0	0
LVAk	13411→0	0	13411	13	0	0	0
PXV	1578→0	0	1492	7	0	4→0	2
GEMET	4→0	894	1	3638	0	3→0	12
SNOMED	102600→0	0	102614	229	0	202→0	0
IPTC	0	0	933	1	0	0	0
NYTP	0	0	4094	0	0	0	6
GTAA	0	0	96850	11894	0	0	0
UNESCO	0	0	2509	227→279	0	0	1524
STW	47→45	25050	5290	10123	214→0	0	0
LCSH	100316→0	0	308607	7766	669→0	206→0	0
AGROVOC	0	32060	29820	2666→2683	0	2424→0	2166
RAMEAU	116343→0	140860→172469	70358	5539→5905	0	33066→0	7940
DDC	0	158161	251977	40729	1→0	0	416
SSW	4→0	1143	1328	39	0	16→0	6
Plant	1→0	0	220	54	0	0	0
DBpedia	0	0	865902	765	0	0	0

RAMEAU uses language tags predominantly with `prefLabel`, `altLabel`, `scopeNote`, and `inScheme` attributes, although the use of the latter does not conform with the SKOS schema (RAMEAU uses a literal instead of a `ConceptScheme` resource as the object of the `inScheme` statement). Furthermore, literals of `dcterms:description` in some cases also have assigned language tags, mostly if the description is given in natural language, e.g., "Suite lithographique pour illustrer l'oeuvre de Shakespeare"@fr but not for position descriptions such as "383-[1] p.". Also, literals of the `dcterms:title` property are sparsely annotated with language tags.

Incomplete Language Coverage is spotted in 11 of the 24 vocabularies. Most concepts in ODT are described with English and German preferred labels, except 16 which lack the German `prefLabel`.

Nearly all of the 6,370 incompletely covered concepts in Eurovoc omit the Irish and Maltese languages (language tags `@ga` and `@mt`); in six cases Hungarian (`@hu`) is missing. Apparently, translation to these languages has not been performed yet, which is reflected by the SKOS-XL[30] labels that state `eu:toBeTranslated` properties with the literals "ga" or "mt" as objects.

AGROVOC contains literals in 25 different languages but 32,060 concepts are not labeled in all languages. Of these, 19 concepts lack labels for only two languages whereas others do not cover up to 24 languages.

STW, which is expressed mainly in English and German, has many concepts with incomplete language coverage because it (i) links to non-authoritative concepts that are only labeled in German and (ii) uses the private, but valid language tag `@x-other` with some of its concept labels.

158,161 concepts in DDC have incomplete descriptions in exactly 13 languages. This happens because concepts are defined separately for different languages. E.g., the concepts `ddc:class/746.44/2007/02/about.it` and `ddc:class/955/2009/03/about.de` each have only an Italian or German `prefLabel` defined. Also, many concepts in DDC only have English labels.

All the 24 vocabularies that we reviewed contained *Undocumented Concepts*. To document concepts, ODT makes heavy use of `definition` properties. However,

[30] SKOS-XL is an extension schema to SKOS that enhances the labeling capabilities by, treating labels as resources and not as literals.

we could find 35 concepts lacking these or other SKOS documentation properties. The most widely used documentation properties in Eurovoc are `scopeNote` but there are 5,341 of 6,797 concepts that remain undocumented. Also all other vocabularies have a significant number of undocumented concepts.

Overlapping Labels were observed in 21 vocabularies. The 765 overlapping labels in DBpedia are caused by duplicate categories which differ only in case, e.g., "Visual Arts" and "Visual arts". ODT shows two cases of label overlap due to the use of the same abbreviations as alternative labels in different concepts.

Abbreviations are a source for overlap also in Eurovoc. For example, the concepts with the preferred label "United Nations High Commissioner for Human Rights" has an alternative label "UNHCR" in Polish language. However, there also exists a concept with an alternative label "United Nations High Commissioner for Refugees" which has been assigned the same abbreviation as the preferred label. Besides these abbreviation-related overlaps we could also observe identical labels being used for different concepts, e.g., "hooldushüvitis"@et defined both as a `prefLabel` for the concept `eurovoc:7946` and as an `altLabel` for the concept `eurovoc:4209`.

In the same way, PXV uses the string "primary peroxisomal enzyme deficiency" with two concepts in the same concept scheme, but once with a `prefLabel` and another time with an `altLabel` property.

Overlapping labels in the UNESCO vocabulary only occur between `prefLabel` values because the other SKOS labeling properties `altLabel` and `hiddenLabel` are not used. The overlap arises because the UNESCO vocabulary is a hierarchical classification where the categories are implicitly qualified by their surrounding context, but the context is not expressed in the label itself. For example, *Theory* appears both under *General demography* and *General sociology*. There are also many categories with the label "Other (specify)"@en.

There are over 10K overlapping labels in STW. They arise because the vocabulary includes mappings to other vocabularies, and the mappings include the labels of the foreign concepts. The current version of qSKOS cannot distinguish between authoritative and non-authoritative concepts when looking for overlapping labels.

Inconsistent Preferred Labels could be found only in 5 out of the 24 reviewed vocabularies. A reason could be that this issue is stated as an integrity condition in the SKOS reference and also covered by thesaurus guidelines [33,17] in a similar way. Thus, vocabulary developers might already check their vocabularies against it.

UMBEL has two inconsistently labeled resources which may be caused by an misunderstanding of the `prefLabel` usage because one of the labels is a longer narrative description of the concept that might be better expressed by using one of the `note` properties.

The only occurence of this issue in GeoNames is caused by inconsistent usage of upper/lowercase in `prefLabel` literals: one concept has both the labels "language school" and "Language School".

All inconsistently labeled resources of STW are resources from DBpedia that are assigned multiple German `prefLabel`s within the STW vocabulary. For example, the resource `dbpedia:Agritourism` has two labels, "Turismo rural"@de and "Agrotourismus"@de.

Inconsistent labels also occur in a greater quantity in LCSH, mostly with minor differences in labeling. The same concept is, e.g., labeled with the `prefLabel`s "Nation-state–Congresses", "National state–Congresses" and "National-state–Congresses".

Compared to the total numbers of concepts in the vocabularies, *Disjoint Labels Violations* seem to be a minor issue that is already handled well by the vocabulary developers. A higher number of occurences of this issue can be found in RAMEAU (>30,000) and AGROVOC (>2,400). All other vocabularies show up to approximately 200 occurencies which is an amount that can be handled by manual correction.

ODT and UMBEL each have one concept labeled identically as `prefLabel` and `altLabel`. The same pattern can be observed with EARTH, SNOMED, AGROVOC, and RAMEAU which also do not make use of `hiddenLabel`s.

Extra Whitespace in Labels occurs in half of the reviewed vocabularies, according to the *Skosify* tool that was used to measure this issue.

5.2.3 Structural Issues

Table 7 summarizes our findings regarding the structure of the vocabularies in our dataset.

Orphan Concepts occur in 17 of the 24 vocabularies. In the GeoNames, NYTL and NYTP vocabularies, all concepts are orphan concepts, which means that these vocabularies are authority files rather than thesauri or taxonomies. This also implies that these vocabularies have no disconnected concept clusters. GTAA is a mixture of name authority file (approx. 162K concepts) and thesaurus (approx. 10K concepts). The 70 orphan concepts in STW are deprecated concepts and marked as such with the `historyNote` property.

All four orphan concepts of ODT are top concepts of the same resource (`odt:Regions`) but not used with any `semanticRelation`s in the vocabulary. These concepts may be very infrequently used which could also be indicated by the so far uncorrected typing error in the preferred label "Ocenania"@en of `odt:Ocenania`. Similarly,

Table 7 Validation and correction results using the *qSKOS* quality analysis toolkit, part 2: *Structural Issues*

	Orphan Concepts	Disconnected Concept Clusters	Cyclic Hierarchical Relations	Valueless Associative Relations	Solely Transitively Related Concepts	Omitted Top Concepts	Top Concepts Having Broader Concepts	Unidirectionally Related Concepts	Relation Clashes	Mapping Clashes
ODT	4	7	0	7→6	0	0	2	126→0	0	0
Eurovoc	7	4	0	6→5	0	1→0	0	14289→0	0	0
UMBEL	2936	86	5→0	0	36535→0	0	0	740→0	0	0
GeoNames	680	0	0	0	0	9→0	0	0	0	0
NYTL	1920	0	0	0	0	1→0	0	0	0	0
EARTh	2288	354	0	1124	0	0	0	12091→0	61→0	0
Reegle	4	2	0	2013→1287	842→0	1	0	1718→0	317→0	2
IPSV	0	1	0	253	0	0	0	25→0	5→0	0
LVAk	21	11	5→0	5	0	0	0	16344→0	1→0	0
PXV	2	10	0	0	0	0	1	2725→0	2→0	0
GEMET	0	5	0	31	0	1→0	0	9657→0	2→0	0
SNOMED	0	1	0	119→115	0	0	0	60396→0	1234→0	0
IPTC	0	10	0	0	1113→0	0	0	2241→0	0	0
NYTP	4979	0	0	0	0	1→0	0	0	0	0
GTAA	162000	621	0	9448→9414	0	9→0	0	18804→0	37→0	0
UNESCO	0	1	0	19	0	0	0	124→0	0	0
STW	70	141	0	5004→5000	0	2	0	18533→0	5→0	0
LCSH	173149	22343	0	0	0	1→0	0	96533→0	0	0
AGROVOC	0	234	0	281	0	0	0	20672→0	1→0	0
RAMEAU	86137	24927	4→0	5118→5037	0	0	0	322079→0	337→0	0
DDC	97294	2087	0	0	0	30→5	1812	4761→0	0	0
SSW	6	1	0	118→46	22→0	0	0	723→0	4→0	0
Plant	0	22	0	3463	0	0	44	3246→0	0	0
DBpedia	103877→103880	1174→1171	1133	9021→6352	0	0	0	1713339→0	10219→0	0

all seven orphan concepts of Eurovoc are top concepts that do not participate in any `semanticRelation`.

The large number of orphan concepts in DDC are caused by the way different versions of a concept are organized. For example, the orphan concept `ddc:class/2--499/e23/` is only related to its versioned counterparts, e.g., `ddc:class/2--499/e23/2012-08-08/`, by the property `dct:hasVersion`. These versioned concepts are then organized in a hierarchical structure.

Disconnected Concept Clusters (DCCs) are found in 21 vocabularies. Three vocabularies show no DCCs because all concepts are orphan concepts and thus no relations between them are established. Four vocabularies (IPSV, SNOMED, UNESCO and SSW) consist of only one "giant component", which is often considered the ideal vocabulary structure.

STW forms one giant component (containing 24,572 concepts), but has also 140 additional DCCs, which all consist of authoritative concepts mapped to third-party vocabularies. All other vocabularies split into several clusters of semantically related concepts, each of which represents a certain subtopic.

Eurovoc has four DCCs, consisting of 6775, 6, 5, and 4 concepts. In the large DCC (the "main" cluster) it uses a custom ontology to organize numerous micro-thesauri and domains and cross-connects concepts by `related` properties. However, this is not the case for the three small DCCs, indicating a quality flaw.

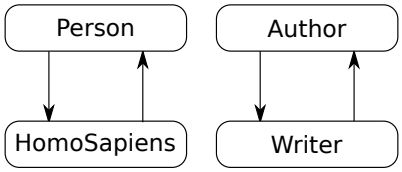
GTAA consists of 621 highly unbalanced DCCs. One component contains 8,413 subjects from a thesaurus with carefully curated semantic relations. Most other components contain fewer than 10 entities from other categories, e.g., locations, person names, and genres.

PXV consists of ten topic-related DCCs, such as "deficiencies", "defects" or "signals". Some of the eleven concept clusters contained in the LVAk thesaurus are obviously forgotten test data.

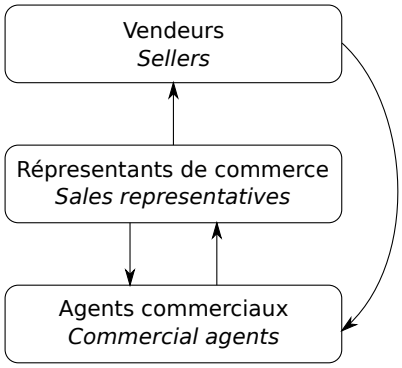
Only four vocabularies contain *Cyclic Hierarchical Relations* which is a comparatively small number. Also, the number of cycles within the vocabularies is small (4 or 5), except for DBpedia, which contains 1,133 cycles.

Four of the five cycles in UMBEL involve only two concepts. Two of these cycles are illustrated in Figure 4(a). One cycle involves three concepts. Also in LVAk the cycles are rather small with five involved concepts at maximum. RAMEAU has one cycle involving 20 concepts; the other three cycles, one of which is illustrated in Figure 4(b), contain only 2–3 concepts.

In the collaboratively created DBpedia vocabulary, many cycles are caused by concepts that have reflexive `broader` relations. The DBpedia authors are aware of this, noting that the "categories do not form a proper



(a) Simple cycles in UMBEL



(b) Complex cycles in RAMEAU

Fig. 4 Examples of simple cycles in UMBEL and a more complex cycle in RAMEAU. English equivalents for French labels shown in *oblique text*.

topical hierarchy, as there are cycles in the category system and as categories often only represent a rather loose relatedness between articles" [9].

The cycles in LVAk could, in our opinion, be resolved by replacing hierarchical with associative relations or synonym definitions.

Valueless Associative Relations have been detected in 16 vocabularies. Some of the potentially valueless associative relations could possibly be fixed by reconsidering the structure and replacing some associative relations by hierarchical ones. This could be observed, e.g., in LVAk and GEMET. The latter definies the concept labeled "leukaemia"@en as `related` to the concept labeled "cancer"@en with a common parent labeled "human disease"@en. Here a hierarchical structure might be worth considering.

In general, the total number of occurrencies of this issue is relatively low compared to the number of all semantic relations in the respective vocabularies. However, revising thousands of relations is still unmanagable for a single thesaurus manager (cf. Section 8).

Solely Transitively Related Concepts were found in four vocabularies. UMBEL only uses `broaderTransitive` and `narrowerTransitive` properties and completely omits `broader` and `narrower` properties. IPTC only uses `broaderTransitive` relations to create a hierarchical structure.

The other two vocabularies having this issue are SSW and Reegle with 22 and 842 occurencies, respectively. Both vocabularies were developed using the *PoolParty Thesaurus Manager* which can be configured to automatically infer `broaderTransitive` and `narrowerTransitive` relations and include them in the vocabulary. Speaking to the developers of the PoolParty system, we were informed that this functionality is now discontinued. However, the exact causes of these "superfluous" transitive relations remain to be investigated.

Omitted Top Concepts were found in 10 of the 24 reviewed vocabularies. NYTL, NYTP, LCSH, GEMET, GTAA, and GeoNames omit top concepts in all the concept schemes they define. Eurovoc uses 128 concept schemes but has one without a top concept, which simply contains all concepts defined in the vocabulary. Such an "umbrella concept scheme" without a top concept is also present in LCSH, NYTL, NYTP, and GEMET. The only concept scheme in Reegle that omits a top concept is automatically created by the PoolParty application and does not contain any concepts. The two omitted top concepts in STW are introduced by the AGROVOC and GESIS[31] mapping files: both of them assign concepts from their originating vocabulary to a concept scheme also in this vocabulary which seem to be "copied" statements from the original publication.

In our selection of vocabularies, only four vocabularies feature *Top Concepts Having Broader Concepts*. ODT defines 29 top concepts, but only two of them have broader concepts. However, the broader concepts of these two concepts are again top concepts.

In its current version, PXV is affected by one top concept that has broader concepts.

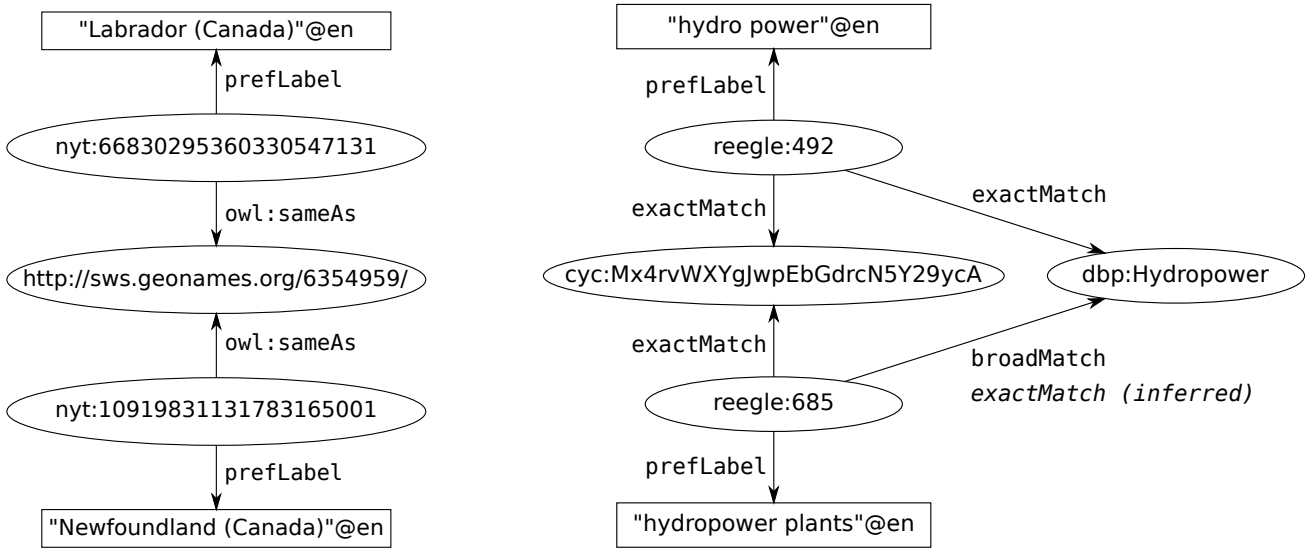
All three concept schemes defined in Plant have associated top concepts. Of these, 44 are related to broader concepts.

Unidirectionally Related Concepts are contained in all except three vocabularies (GeoNames, NYTP, NYTL) which assert the complete set of reciprocal relations.

Relation Clashes occur in 13 of the 24 reviewed vocabularies. We could observe that the associative relations span various hierarchy levels. For LVAk and PXV the maximimum level is one, i.e., concepts that are connected by `related` are also directly connected by `broader`. However, there are also occurrences over multiple levels that are harder to spot like those we observed in Reegle and IPSV, spanning three or four hierarchy levels. The highest number of hierarchy levels that were connected by associative relations were found in SNOMED (7), RAMEAU (26), and DBpedia (38).

qSKOS could find *Mapping Clashes* only in the Reegle vocabulary, where two clashes could be detected.

[31] TheSoz Thesaurus for the Social Sciences, `http://datahub.io/dataset/gesis-thesoz`



(a) Inferred inconsistent labels in NYTL (b) Inferred mapping clashes in Reegle

Fig. 5 Examples of label and mapping clashes caused via OWL inference. The inconsistent labeling in (a) only appears when `owl:sameAs` inference is performed. The clash between `exactMatch` and `broadMatch` mappings in (b) only appears after all possible `exactMatch` relationships are inferred through transitive and symmetric OWL property inference.

They were caused by mappings to GEMET and DBpedia. One of the clashes is illustrated in Figure 5(b).

5.2.4 Linked Data Specific Issues

In Table 8 we give an overview about issues we consider relevant for online publication and interoperability with other vocabularies. We did not include figures of *Missing In-links* and *Broken Links* for LVAk because this vocabulary is not yet published online.

However, except GeoNames and GEMET no vocabulary has a high number of estimated *in-links* from other web resources.

The difference between the number of concepts and the number of authoritative concepts in Table 3 already indicates which vocabularies contain *out-links* to other SKOS vocabularies. Closer examination shows that every authoritative concept in NYTL, NYTP, and Plant is linked to other resources on the Web. UMBEL and SNOMED are also reported to define an outlink for every concept, but this is caused by multiple type definitions (e.g., every concept in UMBEL is also explictly typed as `owl:NamedIndividual` and `owl:Class`), and should be considered in future versions of the tool. In a similar way, DDC defines most concepts as being of type `owl:Thing`. However, due to the large number of mappings to other resources, e.g., RAMEAU, AGROVOC, STW, and GEMET expose a significant difference in the number of authoritative concepts and missing out-links, i.e., many defined concepts reference related third-party resources on the Web.

Table 8 Validation results using the *qSKOS* quality analysis toolkit, part 3: *Linked Data Specific Issues*. Values marked with an asterisk (*) have been extrapolated from a randomly sampled subset of the concepts.

	Missing In-links	Missing Out-links	Broken Links	Undefined SKOS Resources	HTTP URI Scheme Violation
ODT	111	31	37	1	0
Eurovoc	6170*	6797	120790*	0	0
UMBEL	26110*	0	130*	0	0
GeoNames	24	680	11	0	0
NYTL	1892*	0	1376*	0	0
EARTh	14349	9558	410	0	0
Reegle	1447	809	321	1	9
IPSV	4731	4732	1	1	0
LVAk		13411		0	0
PXV	1686	1046	107	0	0
GEMET	3290*	584	40*	0	0
SNOMED	102610*	0	5*	0	0
IPTC	2061	933→2061	2	1	0
NYTP	4965	0	9	0	0
GTAA	171990*	171991	740*	0	0
UNESCO	2509	2509	1	0	0
STW	6781	1463	504	0	0
LCSH	408920*	347560	2640*	0	0
AGROVOC	31680*	17286	160*	0	0
RAMEAU	207260*	34803	132333*	0	0
DDC	250790*	458	110*	0	0
SSW	1941	1606	285	1	1→4
Plant	3246	0	662	0	0
DBpedia	865566*	865902	11400*	0	0

Even though we could not determine the exact number of *Broken Links* because of the large number of links to resolve (over 400K in Eurovoc, over 500K in LCSH), we found that broken links are a common issue in most vocabularies. However, some vocabularies (IPSV, UNESCO, IPTC) contain very few links that could not be dereferenced at the time of testing. For others, e.g., Eurovoc, we were not able to dereference one third of all HTTP URIs mentioned in the vocabulary, including authoritative concepts. This was possibly caused by a misconfiguration of the vocabulary data server.

We were able to spot *Undefined SKOS Resources* in five vocabularies. IPSV uses the deprecated `prefSymbol` property. ODT, Reegle, and SSW still contain the deprecated `subject` property. IPTC states top concepts using the property `HasTopConcept`, which does not match the property definition in the SKOS ontology.

5.2.5 Adherence to SKOS Integrity Conditions

The SKOS integrity conditions S14, S13, S27, and S46 correspond to the *qSKOS* tests for *Inconsistent Preferred Labels*, *Disjoint Labels Violation*, *Relation Clashes*, and *Mapping Clashes*, respectively (cf. Table 4). The SKOS integrity conditions S9 and S37, related to disjoint classes, are not checked in the current version of *qSKOS*. According to the *qSKOS* results shown in Tables 6 and 7, 18 of the 24 vocabularies (75%) have one or more issues that violate the SKOS integrity conditions. Eurovoc, NYTL, IPTC, NYTP, UNESCO, and Plant stand out by not violating any of the integrity conditions tested by *qSKOS*.

6 Correcting Problems

We developed correction heuristics for 12 of the 26 quality issues defined in Section 4, as shown in the last column of Table 4. These corrections and the result of applying them for our test vocabularies are described in this section.

6.1 Correction Heuristics

In the following subsections, we describe the heuristics we have developed to correct some typical, recurring problems in SKOS vocabularies.

6.1.1 Omitted or Invalid Language Tags

Language tags can be added for human-readable labels and documentation properties if the language of the vocabulary is otherwise known. *Skosify* accepts a *default language* parameter which can be used to specify the implicitly known language of untagged literals. However, this approach only works when the language of untagged literals is known and different languages have not been mixed.

6.1.2 Missing Labels

Missing labels for concepts and concept schemes cannot be corrected without adding more information, in the form of documentation triples, to the vocabulary. However, the most basic case, where a SKOS vocabulary contains a single unlabeled concept scheme (or no concept scheme at all, in which case *Skosify* will create one), can be addressed by labeling the concept scheme. This can be done using the *concept scheme label* parameter in *Skosify*. However, *Skosify* does not detect or attempt to correct unlabeled concepts.

6.1.3 Inconsistent Preferred Labels

When a concept has several `prefLabel` values with the same language tag, one of the labels can be selected as the real `prefLabel` value while the rest are converted into `altLabel` values. By default, *Skosify* will retain the *shortest* label, but other options are available for choosing the *longest* label or not performing any correction at all.

6.1.4 Disjoint Labels Violation

When a concept is linked to a label using two different label properties that are defined as disjoint by the SKOS specification, we remove the value for the less important property (`hiddenLabel` < `altLabel` < `prefLabel`). An example of this correction is shown in Figure 3(a).

6.1.5 Extra Whitespace in Labels

Surrounding whitespace from SKOS label or documentation properties can be removed. This correction is performed in *Skosify* before the correction for *Overlapping Labels*, because it may help uncover cases of label overlap that would otherwise remain undetected due to differences in the amount of surrounding whitespace.

6.1.6 Cyclic Hierarchical Relations

We use a naïve approach to detect and optionally remove cycles in hierarchical relations by performing a depth-first search starting from the topmost concepts in the hierarchy. The depth-first search approach for eliminating cycles is simple, fast, and domain independent,

but may not produce deterministic results and "cannot ensure that the links ignored during the graph traversal in order to prevent loops from happening are actually the appropriate links to be removed" [30]. More accurate formal methods for eliminating cycles in terminological hierarchies exist, but they are more complex and not as general as the naïve approach [30].

6.1.7 Solely Transitively Related Concepts

To eliminate `broaderTransitive` and `narrowerTransitive` relationships that cannot be inferred from the asserted hierarchy, we first remove all transitive hierarchical relations from the vocabulary and then optionally recreate them from the asserted `broader` and `narrower` relationships. This ensures that the inferred transitive relationships match the explicitly asserted hierarchy.

6.1.8 Omitted Top Concepts

While this quality issue is not specifically targeted by *Skosify*, it is often at least partially resolved by the correction heuristic for *Unmarked Top Concepts*, described below. Explicitly marking top concepts using `hasTopConcept` and `topConceptOf` relations makes it less likely that a concept scheme will remain without top concepts.

6.1.9 Unmarked Top Concepts

We ensure that top concepts are explicitly marked as such using a three-step process: (i) if the vocabulary does not contain any concept schemes, we create one; (ii) we infer the concept scheme for every concept that is not marked as belonging to a concept scheme with the `inScheme` property by selecting, when necessary, one concept scheme as the *default* concept scheme for a vocabulary[32]; and (iii) for each concept scheme, we identify the top level concepts in that concept scheme (i.e., the concepts having no `broader` relationships) and add `hasTopConcept` and `topConceptOf` relationships between the concept and its concept scheme.

6.1.10 Unidirectionally Related Concepts

We enrich the SKOS vocabulary with bidirectional relationships when possible, i.e., infer `related` relationships for both directions and also infer the inverse relationships for `broader` and `narrower`. We perform a similar enrichment for the corresponding mapping relationships `relatedMatch`, `broadMatch` and `narrowMatch`. An option in *Skosify* makes it possible to instead omit the `narrower` relationships because they can be considered redundant in some scenarios.

6.1.11 Relation Clashes

We address the combined use of relationships that are defined as disjoint by the SKOS specification by removing the less important relationship. In particular, the `related` relationship is often used to link between concepts that are directly above or below each other in the `broader` hierarchy, as shown in Figure 3(b) and 3(c). In this situation, we remove the `related` relationship assertion, leaving the `broader` hierarchy intact. This correction is performed by default in *Skosify*, in order to enforce the SKOS integrity condition S27, but can be optionally disabled.

6.1.12 Disjoint Classes Violation

Some relationships intended for `Concept`s, such as the mapping relationship `exactMatch`, were found to be used on `Collection` instances in some vocabularies we analyzed in our earlier study [41]. The RDFS inference capabilities of the *PoolParty checker* together with `rdfs:domain` specifications of some SKOS properties caused those instances to be marked both as `Concept`s and `Collection`s. We identify this particular error in *Skosify* and correct it by removing the improper relationship assertions. However, *Skosify* cannot correct the more general case where a resource is explicitly marked as being of several types that are defined to be disjoint.

6.1.13 Other Corrections

We have also implemented a generic property and class substitution mechanism in *Skosify*, which may be used to convert specific properties into a new property or instances of a specific class into instances of another class. This mechanism was originally developed to facilitate the conversion of non-SKOS RDF vocabularies, such as lightweight OWL ontologies, into SKOS. For example, a lightweight OWL ontology may be converted into simple SKOS format by converting instances of `owl:Class` into instances of `Concept`, `rdfs:subClassOf` relationships to `broader`, and `rdfs:label` properties to `prefLabel`. This mechanism may also be used to correct misspellings and other similar problems where an invalid property or class is used.

Skosify also optionally supports simple RDFS subclass and sub-property inference, which will be performed before correction heuristics are applied. It can

[32] In the most common case, there is only one concept scheme (often the one created in the previous step), and that will be selected as the default concept scheme; otherwise, the default concept scheme will be chosen arbitrarily and a warning message shown by *Skosify*.

be used when a vocabulary specializes SKOS by defining its own constructs as sub-properties or sub-classes of SKOS constructs.

6.2 Correction Settings

We determined the optimal *Skosify* settings for correcting each vocabulary as follows:

1. The *default language* setting was used when the vocabulary was found to be missing language tags and a manual inspection found that the literals without language tags were unambiguously in a specific language.
2. A *concept scheme label* was set when the vocabulary did not contain a labeled `ConceptScheme` instance. In these cases, a *default language* setting was also used, as it is used as the language tag for the concept scheme label.
3. Breaking of cycles in the hierarchy was enabled if the vocabulary was found to contain cycles, except in the case of DBpedia Categories, where we considered the numerous cycles to be an intrinsic feature of the vocabulary, possibly carrying meaning that would be lost if the cycles were broken.
4. RDFS inference was enabled if the vocabulary contained sub-class or sub-property axioms involving SKOS constructs.
5. For IPTC and UMBEL, the generic property mapping functionality in *Skosify* was used to replace `broaderTransitive` relationships in the original vocabulary with `broader`. For UMBEL, `narrowerTransitive` was similarly replaced with `narrower`.
6. For IPTC, we also used the property mapping functionality to correct some invalid namespaces and misspellings that were present in the original file.

The settings we used for each vocabulary are summarized in Table 9. For all other *Skosify* settings, we used the default values: `narrower` relationships were created when necessary, `related` relationships violating the SKOS integrity condition S27 were eliminated, and in the case of inconsistent `prefLabel` values, the *shortest* label was retained. Transitive hierarchical relationships were not generated.

6.3 Correction Results

After processing each vocabulary with *Skosify* using the correction settings discussed above, we reanalyzed them using both the *PoolParty checker* tool and the *qSKOS* tool.

Table 9 *Skosify* correction settings used for each vocabulary.

	Default language	Concept scheme label	Break cycles	RDFS inference	Other settings
ODT	en	-	-	-	
Eurovoc	-	-	-	-	
UMBEL	en	X	X	X	`broaderTransitive`→`broader`; `narrowerTransitive`→`narrower`.
GeoNames	-	-	-	X	
NYTL	en	X	-	-	
EARTh	en	-	-	-	
Reegle	en	-	-	-	
IPSV	en	X	-	-	
LVAk	de	X	X	-	
PXV	en	X	-	-	
GEMET	-	-	-	-	
SNOMED	fr	X	-	-	
IPTC	en	-	-	-	`broaderTransitive`→`broader`; fix invalid SKOS namespace; fix misspelled `hasTopConcept`.
NYTP	en	X	-	-	
GTAA	-	-	-	-	
UNESCO	-	-	-	-	
STW	de	-	-	X	
LCSH	en	X	-	-	
AGROVOC	en	X	-	-	
RAMEAU	fr	X	X	-	
DDC	-	-	-	-	
SSW	en	-	-	-	
Plant	en	-	-	-	
DBpedia	en	X	-	-	

6.3.1 Correction Results according to the PoolParty checker

In the initial evaluation of 17 vocabularies performed using the *PoolParty checker* and described in Section 5.1, 13 (76%) of the test vocabularies were found to be inconsistent with the SKOS integrity conditions and also many vocabularies had issues with missing language tags, missing labels, and loose concepts. After processing these vocabularies using the *Skosify* tool, many of these problems were eliminated. The results of reanalyzing the vocabularies are shown in Table 5, with the value after *Skosify* processing shown after the arrow symbols.

Of the 13 vocabularies that failed one or more of the mandatory checks, 11 were succesfully corrected so they subsequently passed all the mandatory checks. The issues regarding *Consistent Use of Semantic Relations* were all succesfully corrected, while the *Consistent Use of Labels* issues were corrected in all but two vocabularies. In NYTL and Reegle, the inconsistency in preferred labels only arises when `owl:sameAs` inference is taken into account, i.e., two concepts with different URIs, but with an `owl:sameAs` relationship, use different labels in the same language. An example of this is illustrated in

Figure 5(a). Similarly, the Reegle vocabulary violates the SKOS integrity conditions for mapping properties (*Consistent Use of Mapping Relations*) when OWL inference is performed to infer all possible relationships, as illustrated in Figure 5(b). These problems could not be addressed by the correction heuristics implemented in *Skosify*.

All the issues involving *Missing Language Tags* and *Loose Concepts* were succesfully corrected by *Skosify*. The *Missing Labels* issues could not be corrected, as in most cases the problem was the lack of labeling information for concepts and/or concept schemes, and *Skosify* was unable to add the missing information. For UMBEL and GeoNames it appeared that new problems with *Missing Labels* were caused by *Skosify* processing. However, the unlabeled resources causing the check to fail were already present in the original vocabulary, but the *PoolParty checker* apparently failed to recognize them as `Concept` and `ConceptScheme` instances until after the RDFS inferences performed by *Skosify* had explicitly set their types.

6.3.2 Correction Results according to qSKOS

In the *qSKOS* quality analysis, it was discovered that 18 of the 24 analyzed vocabularies (75%) violate one or more of the SKOS integrity conditions checked by *qSKOS* (cf. Section 5.2.5). The *qSKOS* analysis also discovered many issues with missing or invalid language tags as well as problems involving cyclic hierarchical relationships, invalid use of transitive hierarchical relationships, omitted top concepts, and lack of bidirectional relationships. As with the *PoolParty checker* evaluation described above, most of the problems were successfully eliminated with the *Skosify* processing. The results of reanalyzing the vocabularies are shown in Tables 6 and 7, with the value after *Skosify* processing shown after the arrow symbols.

Of the 18 vocabularies that violated one or more of the SKOS integrity conditions according to *qSKOS*, all vocabularies except Reegle were successfully corrected so that a subsequent analysis found no remaining issues involving the integrity conditions. In Reegle, the two *Mapping Clashes* remained, as in the *PoolParty checker* results.

Omitted or Invalid Language Tags were corrected in all vocabularies except for Eurovoc and STW. In Eurovoc, a default language setting was not used as the untagged literals were mostly country codes, for which no suitable language tag could be assigned. In STW, the 45 `@x-other` language tags could not be corrected. *Cyclic Hierarchical Relations* were eliminated in the three vocabularies for which the corresponding setting was enabled; in DBpedia, we chose not to remove the large number of cycles. *Solely Transitively Related Concepts* were corrected in all four affected vocabularies. *Omitted Top Concepts* were successfully corrected in most vocabularies, though some omitted top concepts remained in Reegle, STW and DDC. Finally, *Unidirectionally Related Concepts* issues were corrected in all 21 affected vocabularies.

The *Skosify* processing also affected some quality measures that were not explicitly targeted by the correction heuristics. The *Incomplete Language Coverage* value increased for RAMEAU, because the assignment of language tags by *Skosify* caused *qSKOS* to examine many concepts that previously didn't have any language tags in their labels and had therefore been bypassed in the original check.

There was a moderate increase in the number of *Overlapping Labels* in five vocabularies. The increase is due to the stripping of extra whitespace in labels performed by *Skosify*. This normalization of labels increases the recall of the *qSKOS* check for overlapping labels, which is sensitive to whitespace.

In DBpedia, the number of *Orphan Concepts* increased by three and the number of *Disconnected Concept Clusters* decreased by the same number. The affected three concepts all had `related` relationships to themselves. These relationships were eliminated by the correction heuristic which aims to correct *Relation Clashes*, causing *qSKOS* to classify them as orphan concepts instead of separate clusters. Similarly, the number of *Valueless Associative Relations* decreased in nine vocabularies as a side effect of removing `related` relationships violating the SKOS integrity condition S27.

The number of *Missing Out-links* in IPTC more than doubled after the *Skosify* processing. The increase is due to the SKOS inferences performed by *Skosify*, which caused *qSKOS* to examine a larger set of concepts. Similarly, the number of *HTTP URI Scheme Violations* increased in SSW due to the SKOS inferences performed by *Skosify*. The non-HTTP URIs were already mentioned in the original vocabulary, but unnoticed by *qSKOS*.

7 Discussion and Conclusions

7.1 Defining Quality Issues

The first research question in this study was:

> *How can the quality of SKOS vocabularies be automatically measured?*

To answer this question, we formulated a set of 26 quality issues for SKOS vocabularies that highlight

possible problems in vocabulary quality. Our quality issues are based on earlier literature, discussions with vocabulary users, and manual analysis of published Web vocabularies as well as existing methods for vocabulary evaluation including the SKOS integrity conditions and tools such as the *PoolParty checker*. We then implemented checks for these issues in the *qSKOS* and *Skosify* tools.

The wide adoption of SKOS for publishing controlled vocabularies has enabled and facilitated the use of quantitative quality measures to assess and evaluate quality aspects of vocabularies and to compare different vocabularies with respect to their quality. In contrast to Kless and Milton's abstract evaluation measures for thesauri [24], we have contributed a set of formal, computable quality measures and provided tools that implement them. Our quality measures concentrate on appropriate and correct representation of vocabulary constructs, not on the OWL modeling aspects of SKOS vocabularies as in Abdul Manaf's work [2].

7.2 Observed Quality Issues in Vocabularies

The second research question in this study was:

> *To what extent are existing SKOS vocabularies on the Web affected by quality problems?*

To answer this question, we first collected a representative set of 24 SKOS vocabularies in different domains and size categories. We then converted the vocabularies into a uniform format, manually correcting syntactic or representation-related issues when necessary. We analyzed the vocabularies using the *qSKOS*, *Skosify* and *PoolParty checker* tools to evaluate the vocabularies and look for quality issues.

We found possible quality issues in all of the analyzed vocabularies, in line with findings of our earlier studies of SKOS vocabulary quality [41,26]. All vocabularies had a number of undocumented concepts. A majority of vocabularies included orphan concepts and many contained clusters of concepts unconnected to the main cluster.

A particularly worrying finding was that in both the *qSKOS* and *PoolParty checker* analysis, around three quarters of the analyzed vocabularies were found to violate the SKOS integrity conditions. This may not be surprising, considering that RDF data published online has been found to contain many errors in previous studies [13,18,19]. However, earlier studies did not look specifically at the validity of SKOS vocabularies, with the exception of Abdul Manaf's work [2] that only considered a small number of OWL modeling issues.

We found that the SKOS integrity condition S27, which specifies that the `related` relationship is disjoint with the `broaderTransitive` relationship, is violated by the majority of the vocabularies we examined. In some cases, such as the complex hierarchies in LCSH (cf. Figure 3(c)), the invalid `related` relationships bridge many levels of the concept hierarchy, and may be considered useful for users in navigating the vocabulary to find suitable concepts. Thus, the integrity condition in its current form could be considered overly strict. It could be amended by only forbidding `related` relationships between direct descendants, i.e., specifying that `related` is only disjoint with `broader`, not the transitive variant.

7.3 Correcting Problems in Vocabularies

The third research question in this study was:

> *Can the quality of SKOS vocabularies be improved using an automated process?*

To answer this question, we developed a set of correction heuristics for the subset of quality issues where we considered automatic or semi-automatic correction to be feasible. We implemented these heuristics in the *Skosify* tool. We chose the optimal correction settings for each vocabulary based on the detected quality issues, iterating when necessary. The correction settings were all such that human judgement was necessary to determine which settings to use; e.g., missing concept scheme labels cannot be guessed by *Skosify* because the data is simply not included in the vocabulary. Settings such as cycle breaking, RDFS inference, or any relationship substitutions must take into account the intended use of the vocabulary.

After applying the heuristics to the test vocabularies using *Skosify* and reanalyzing them with the *PoolParty checker* and *qSKOS* tools, we found that most quality issues in the original vocabularies had indeed been corrected. However, some complex problems, especially situations involving inference, were not corrected by our methods. Some of the corrections caused loss of vocabulary data, in particular the elimination of cycles, disjoint labels and `related` relationships that violated the SKOS integrity condition S27. However, these corrections are all optional in the *Skosify* tool, so they can be turned off if desired.

The *qSKOS* and *PoolParty checker* tools give somewhat different results for the success in the correction. According to *qSKOS*, all the issues related to SKOS integrity conditions in all vocabularies were corrected, but the *PoolParty checker* indicated some remaining

issues in NYTL and Reegle. The difference is due to the lack of RDFS and OWL inference in the *qSKOS* tool; the remaining issues only manifested themselves when inferencing was performed.

7.4 Recommendations for Best Practices

Although there are many tutorials for creating and publishing SKOS vocabularies, such as the SKOS Primer [22], there are some aspects of the publishing that could benefit from more explicitly specified best practices. In particular, the question of what relationships to explicitly assert in the published vocabulary and what to leave for the vocabulary user to infer is not always clear. All SKOS semantic relationships between concepts are either symmetric (e.g., `related` and `exactMatch`) or have an inverse counterpart (e.g., `broader` and `narrower`, and their transitive and mapping variants). In principle, a rather small set of relationships can be used to specify the whole vocabulary, and the remaining (redundant) ones inferred using RDFS and OWL inference. This may be a good strategy for editing SKOS vocabularies: minimal assertions are used during editing, and the rest are inferred only when publishing the vocabulary. The inference can be performed by a tool such as *Skosify*. This way, some inconsistent assertions involving inferred relationships, such as the instances of *Solely Transitively Related Concepts* we found in some vocabularies, can be avoided.

In practice, inference is not always possible or desirable for vocabulary users. Applications making use of SKOS vocabularies may benefit from explicitly asserted relations, even if they are in principle redundant and could have been inferred. We thus propose the following guidelines for the inclusion of SKOS relationships in vocabularies published on the Web of Data:

1. Explicitly declare the types of SKOS `Concept`, `ConceptScheme` and `Collection` instances, even if they could be inferred. This is in line with the recommendation by Abdul Manaf et al. [2].
2. Include one or more concept schemes describing your vocabulary and label them appropriately. Assert the full set of both `topConceptOf` and `hasTopConcept` relationships. Make sure `inScheme` relationships are asserted for every concept.
3. Assert the full set of both `broader` and `narrower` relationships. This is also in line with the recommendation by Abdul Manaf et al. [2]. However, do not include the `broaderTransitive` and `narrowerTransitive` relationships, as they are only likely to be useful in special scenarios, may add a lot of new assertions to the vocabulary, and may be inferred by the vocabulary user when necessary.
4. Assert `related` properties both ways.
5. Assert mapping relationships only one way, with concepts from your own vocabulary as the subjects. This is to avoid "SKOS vocabulary hijacking", i.e., the assertion of facts about vocabularies published by others, which is similar to *ontology hijacking* [18].

7.5 Limitations of Our Approach

Due to our focus on computable, data-oriented quality issues, we leave out more intellectual criteria, such as "appropriate specificity" of the vocabulary and the meaning of semantic relations. Thus, our findings and the automated corrections may be judged by domain experts as inappropriate or even wrong for a specific usage scenario. We believe that our approach reveals its full potential in assisting human experts in their vocabulary development tasks, just as spell checkers do in word processors or code checks are performed in integrated development environments for programming languages.

We also focus on intrinsic quality, i.e., we analyze the vocabularies as isolated entities. In reality a controlled vocabulary is most often used in conjunction with other resources, e.g., a corpus of documents. On the Web, however, it is rarely possible to evaluate the vocabularies in relation with their associated corpus, because it is not available for download or does not (only) cover digital objects. Furthermore, our proposed quality checking functions are designed for short execution times (except *Missing In-links* and *Broken Links* checking, which rely on dereferencing and querying external resources), being applicable on a regular basis.

8 Future Work

In this study, we used three different and complementary tools to analyze and correct SKOS vocabularies. The different tools were originally created separately. From a user point of view it would be beneficial to have a single unified tool which would implement testing for all the quality issues and would also be able to correct problems. It is unlikely that our current tools, *qSKOS* and *Skosify*, could be merged in the near future, due in part to differences in implementation language. However, we are working on expanding the coverage of the tools so that each tool could be used to assess and correct more issues.

The study of our vocabulary dataset showed that *qSKOS* can compute the quality functions in a robust way with good performance and usability of the reports.

However, we also identified various areas of improvement that could lead to more complete and precise analysis reports.

The current test of overlapping labels is case-insensitive, takes into account all SKOS labeling properties, and works across concept schemes. This ensures broad coverage of potential problems, but can lead to false positives in the form of reported issues that do not actually cause real harm. The test could be made configurable with respect to case-sensitivity and the set of labeling properties and concepts to examine. The reporting of problems could be ordered by severity, with, e.g., conflicts between preferred labels reported as more severe than overlap between preferred and alternative labels.

Reports for out-links are currently inaccurate in cases where vocabulary concepts are also instances of types in another namespace (e.g., `owl:Thing`). Future versions of *qSKOS* might consider to exclude the `rdf:type` property from out-link checking or investigate only on a specified set of properties whether they link to external resources.

We are currently working on the integration of qSKOS into existing vocabulary development processes with continuous automated quality checking and feedback to the developers. In subsequent work, we plan to establish such settings and measure the overall impact on the resulting quality of the vocabulary.

The heuristics we implemented in *Skosify* were successfully used to resolve the majority of the targeted quality issues. However, the set of heuristics could be further expanded, e.g., to correct the *Top Concepts Having Broader Concepts* issue. Natural language processing techniques could be incorporated into the correction heuristics in order to, e.g., derive missing language tags for multi-language vocabularies or to add missing labels by examining external sources.

Acknowledgements We thank Eero Hyvönen, Jouni Tuominen, and Miika Alonen for giving insightful comments and support; Andreas Blumauer and Alexander Kreiser for technical assistance with the PoolParty checker; and Andrew Gibson and Tom Dent for providing RDF dumps of the Peroxisome Knowledge Base and the Integrated Public Sector Vocabulary. The work is supported by the FWF P21571 Meketre project[33], the National Semantic Web Ontology project in Finland FinnONTO[34] (2003–2012), and the Linked Data Finland project[35] (2012–2014).

[33] `http://www.meketre.org`

[34] `http://www.seco.tkk.fi/projects/finnonto/`

[35] `http://www.seco.tkk.fi/projects/ldf/`

References

1. ISO 25964-1: Information and documentation – Thesauri and interoperability with other vocabularies – Part 1: Thesauri for information retrieval. Norm, International Organization for Standardization (2011)
2. Abdul Manaf, N.A., Bechhofer, S., Stevens, R.: Common modelling slips in SKOS vocabularies. In: P. Klinov, M. Horridge (eds.) OWLED, *CEUR Workshop Proceedings*, vol. 849. CEUR-WS.org (2012). URL `http://dblp.uni-trier.de/db/conf/owled/owled2012.html#ManafBS12`
3. Abdul Manaf, N.A., Bechhofer, S., Stevens, R.: The current state of SKOS vocabularies on the Web. In: E. Simperl, P. Cimiano, A. Polleres, O. Corcho, V. Presutti (eds.) ESWC, *Lecture Notes in Computer Science*, vol. 7295, pp. 270–284. Springer (2012). URL `http://dblp.uni-trier.de/db/conf/esws/eswc2012.html#ManafBS12`
4. Aitchison, J., Gilchrist, A., Bawden, D.: Thesaurus construction and use: a practical manual. Aslib IMI (2000)
5. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann (2011)
6. van Assem, M., Malaisé, V., Miles, A., Schreiber, G.: A method to convert thesauri to SKOS. In: Y. Sure, J. Domingue (eds.) Proceedings of the Third European Semantic Web Conference (ESWC'06), *Lecture Notes in Computer Science*, vol. 4011, pp. 95–109. Springer-Verlag, Budva and Montenegro (2006). URL `http://www.cs.vu.nl/~mark/papers/Assem06b.pdf`
7. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Computing Surveys **41**(3), 16 (2009)
8. Berrueta, D., Fernández, S., Frade, I.: Cooking http content negotiation with vapour (2008). URL `http://CEUR-WS.org/Vol-368/paper3.pdf`
9. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web **7**(3), 154 – 165 (2009). DOI 10.1016/j.websem.2009.07.002
10. Borst, T., Fingerle, B., Neubert, J., Seiler, A.: How do libraries find their way onto the Semantic Web? Liber Quarterly **19**(3/4) (2010)
11. Byrne, G., Goddard, L.: The strongest link: Libraries and linked data. D-Lib Magazine **16**(11/12) (2010). DOI 10.1045/november2010-byrne
12. de Coronado, S., Wright, L.W., Fragoso, G., Haber, M.W., Hahn-Dantona, E.A., Hartel, F.W., Quan, S.L., Safran, T., Thomas, N., Whiteman, L.: The NCI Thesaurus quality assurance life cycle. J. Biomed. Inform. **42**(3), 530–539 (2009)
13. Ding, L., Finin, T.: Characterizing the semantic web on the web. Electrical Engineering **4273**(August), 5–9 (2006)
14. Fürber, C., Hepp, M.: Using semantic web resources for data quality management. In: Proceedings of the 17th international conference on Knowledge engineering and management by the masses, EKAW'10, pp. 211–225. Springer-Verlag, Berlin, Heidelberg (2010). URL `http://dl.acm.org/citation.cfm?id=1948294.1948316`
15. Harpring, P.: Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works. Getty Publications, Los Angeles (2010)
16. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011). URL `http://linkeddatabook.com/`

17. Hedden, H.: The accidental taxonomist. Information Today (2010)
18. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the pedantic web. In: Proc. WWW2010 Workshop on Linked Data on the Web (LDOW) (2010)
19. Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of Linked Data conformance. Web Semantics: Science, Services and Agents on the World Wide Web **14**, 14–44 (2012)
20. Hopcroft, J.E., Tarjan, R.E.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6), 372–378 (1973)
21. Horridge, M., Parsia, B., Sattler, U.: Explaining inconsistencies in OWL ontologies. In: Proceedings of the 3rd International Conference on Scalable Uncertainty Management, SUM '09, pp. 124–137. Springer-Verlag, Berlin, Heidelberg (2009). DOI 10.1007/978-3-642-04388-8_11. URL `http://dx.doi.org/10.1007/978-3-642-04388-8_11`
22. Isaac, A., Summers, E.: SKOS Simple Knowledge Organization System Primer. Working Group Note, W3C (2009). URL `http://www.w3.org/TR/skos-primer/`
23. Kalyanpur, A.: Debugging and repair of OWL ontologies. Ph.D. thesis, College Park, MD, USA (2006). AAI3222483
24. Kless, D., Milton, S.: Towards quality measures for evaluating thesauri. In: S. Sánchez-Alonso, I. Athanasiadis (eds.) Metadata and Semantic Research, *Communications in Computer and Information Science*, vol. 108, pp. 312–319. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-16552-8_28. URL `http://dx.doi.org/10.1007/978-3-642-16552-8_28`
25. Mader, C., Haslhofer, B.: Quality criteria for controlled web vocabularies. In: International Conference on Theory and Practice of Digital Libraries 2011, NKOS Workshop. Berlin, Germany (2011). URL `http://eprints.cs.univie.ac.at/2923/`
26. Mader, C., Haslhofer, B., Isaac, A.: Finding quality issues in SKOS vocabularies. In: P. Zaphiris, G. Buchanan, E. Rasmussen, F. Loizides (eds.) Theory and Practice of Digital Libraries, *Lecture Notes in Computer Science*, vol. 7489, pp. 222–233. Springer Berlin Heidelberg (2012). DOI 10.1007/978-3-642-33290-6_25. URL `http://dx.doi.org/10.1007/978-3-642-33290-6_25`
27. Malmsten, M.: Making a library catalogue part of the semantic web. In: Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications, pp. 146–152. Dublin Core Metadata Initiative (2008)
28. Miles, A., Bechhofer, S.: SKOS Simple Knowledge Organization System Reference. Recommendation, W3C (2009). URL `http://www.w3.org/TR/skos-reference/`
29. Miles, A., Rogers, N., Beckett, D.: Migrating thesauri to the semantic web – Guidelines and case studies for generating RDF encodings of existing thesauri. SWAD-Europe project deliverable 8.8, SWAD-Europe (2004). URL `http://www.w3.org/2001/sw/Europe/reports/thes/8.8/`
30. Mougin, F., Bodenreider, O.: Approaches to eliminating cycles in the UMLS Metathesaurus: Naïve vs. formal. In: American Medical Informatics Association (AMIA) Annual Symposium Proceedings, pp. 550–554 (2005)
31. Nagy, H., Pellegrini, T., Mader, C.: Exploring structural differences in thesauri for SKOS-based applications. I-Semantics '11, pp. 187–190. ACM (2011). DOI http://doi.acm.org/10.1145/2063518.2063546
32. Neubert, J.: Bringing the "Thesaurus for Economics" on to the web of linked data. In: Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOW2009) (2009)
33. NISO: ANSI/NISO Z39.19 - Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies (2005)
34. Ovchinnikova, E., Wandmacher, T., Kühnberger, K.: Solving terminological inconsistency problems in ontology design. International Journal of Interoperability in Business Information Systems **2**(1), 65–80 (2007)
35. Pipino, L., Lee, Y., Wang, R.: Data quality assessment. Commun. ACM **45**(4), 211–218 (2002)
36. Popitsch, N.P., Haslhofer, B.: DSNotify: handling broken links in the web of data. In: Proc. 19th Int. Conf. World Wide Web (WWW), pp. 761–770 (2010). DOI 10.1145/1772690.1772768
37. Poveda-Villalón, M., Suárez-Figueroa, M., Gómez-Pérez, A.: Validating ontologies with OOPS! In: A. Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, N. Hernandez (eds.) Knowledge Engineering and Knowledge Management, *Lecture Notes in Computer Science*, vol. 7603, pp. 267–281. Springer Berlin Heidelberg (2012). DOI 10.1007/978-3-642-33876-2_24. URL `http://dx.doi.org/10.1007/978-3-642-33876-2_24`
38. Schandl, T., Blumauer, A.: PoolParty: SKOS thesaurus management utilizing linked data. In: Proceedings of the 7th Extended Semantic Web Conference (ESWC2010) (2010)
39. Soergel, D.: Thesauri and ontologies in digital libraries: tutorial. In: Proc. 2nd Joint Conf. on Digital libraries (JCDL) (2002)
40. Summers, E., Isaac, A., Redding, C., Krech, D.: LCSH, SKOS and Linked Data. In: Proceedings of the International Conference on Dublin Core and Metadata Applications (DC-2008), pp. 25–33. Dublin Core Metadata Initiative (2008)
41. Suominen, O., Hyvönen, E.: Improving the quality of SKOS vocabularies with Skosify. In: Proceedings of the 18th international conference on Knowledge Engineering and Knowledge Management, EKAW'12, pp. 383–397. Springer-Verlag, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-33876-2_34. URL `http://dx.doi.org/10.1007/978-3-642-33876-2_34`
42. Svenonius, E.: Definitional approaches in the design of classification and thesauri and their implications for retrieval and for automatic classification. In: Proc. Int. Study Conference on Classification Research, pp. 12–16 (1997)
43. Tuominen, J., Frosterus, M., Viljanen, K., Hyvönen, E.: ONKI SKOS server for publishing and utilizing SKOS vocabularies and ontologies as services. In: Proceedings of the 6th European Semantic Web Conference (ESWC 2009). Springer-Verlag (2009)
44. Vrandecic, D.: Ontology evaluation. Ph.D. thesis, KIT, Fakultät für Wirtschaftswissenschaften, Karlsruhe (2010)