A Thesaurus of Old English as Linguistic Linked Data: Using OntoLex, SKOS and *lemon-tree* to Bring Topical Thesauri to the Semantic Web

Sander Stolk

Leiden University, Leiden, the Netherlands
E-mail: s.s.stolk@hum.leidenuniv.nl

Abstract

An increasing number of dictionaries are represented on the Web in the form of linguistic linked data, utilizing OntoLex-Lemon for this purpose. Lexicographic resources other than dictionaries, however, have thus far not been the main focus of efforts surrounding this model. In this paper, we discuss porting a topical thesaurus to the Web: *A Thesaurus of Old English*. By means of this case study, this paper discusses how this thesaurus – and topical thesauri in general – can be represented with OntoLex-Lemon, SKOS and *lemon-tree* through a fully automated process. Along with discussing the terminology required for expressing *A Thesaurus of Old English* as linguistic linked data, this paper indicates challenges encountered in the conversion process. These challenges range from material that is not meant to be made available to the general public to distinctions and relations that have been left implicit in the legacy form but are of much value and, indeed, required to be expressed explicitly in its linked data form. The aim of this paper, thus, is to provide recommendations for representing topical thesauri on the Web and to grant insight into aspects that may be encountered in porting similar lexicographic resources in the future.

Keywords: thesaurus; linguistic linked data; conversion; automation

1. Introduction

An increasing number of dictionaries are represented on the Web in the form of linguistic linked data using the OntoLex-Lemon vocabulary (Bosque-Gil et al., 2016; Khan, 2016). Such a representation is thought to facilitate interoperability across linguistic resources, have the potential to increase their visibility, and promote their reuse (Declerck et al., 2015; Klimek & Brümmer 2015). However, lexicographic resources other than dictionaries have thus far not been the main focus of efforts surrounding OntoLex-Lemon and its modules. In this paper, we discuss porting a topical thesaurus to the Semantic Web: *A Thesaurus of Old English.*

A Thesaurus of Old English captures the lexis of the early medieval variant of English, spoken between roughly 500 and 1100 by the Anglo-Saxons (Roberts et al., 2015). This lexicographic resource presents a feature common to topical thesauri but uncommon to dictionaries: its topical system (i.e., a hierarchy of categories) that organizes lexical senses according to their meaning (Kay & Alexander, 2016). Moreover, this thesaurus

also distinguishes conceptual levels within the topical system – a feature that was already present in the first modern thesaurus, *Roget's Thesaurus* (1852). By means of this case study, then, this paper presents areas problematic for representing *A Thesaurus of Old English* – and topical thesauri in general – in OntoLex-Lemon alone, and turns to the novel model *lemon-tree* for the needed expressivity. This model combines OntoLex-Lemon with the SKOS vocabulary, filling minor but important lacunae perceived for topical thesauri specifically, thereby increasing the portability and interoperability of these lexicographic resources (Stolk, 2019).

Next to treating the terminology required for porting *A Thesaurus of Old English* to a linguistic linked data form, this paper will indicate further challenges in this process. These range from material available in the legacy form that is not meant to be made available to the general public (e.g., notes purely editorial in nature) to distinctions and relations that have been left implicit in the legacy form but are of much value and, indeed, required to be expressed explicitly in its linked data form. The aim for this paper, thus, is to provide recommendations for representing topical thesauri on the Web and to grant insight into aspects that may be encountered in porting similar lexicographic resources in the future.

2. *A Thesaurus of Old English*

A Thesaurus of Old English (TOE) captures the lexis of Old English. The words and their senses of this historical variant of English, spoken roughly between 500 and 1100, are grouped together in sets of synonyms and placed in an overarching hierarchy of categories. In addition, TOE indicates the distribution of words in the surviving Old English texts. Thus, some are flagged as found only in poetic works or as glosses. As of May 2017, the thesaurus contains 51,483 senses that have been sorted and categorized manually in 22,451 categories[1]. Accumulating and editing this wealth of information for the first publication of the thesaurus in 1995 took a team of scholars – led by Christian Kay, Jane Roberts, and Lynne Grundy – over fifteen years (Roberts, 1978). The fruit of their labour has certainly not gone unnoticed in the scholarly field concerning Old English.

Since its publication, TOE has been met with high praise. Rolf Bremmer Jr, for instance, states that the thesaurus fills a "voluminous gap [...] on the shelf of lexicographical tools" available for Old English (2002). Richard Dance, too, calls TOE "invaluable" for lexical studies and deems it an "impressive piece of scholarship" (1997). Manfred Görlach goes so far as to state that TOE is "the most important contribution to Old English studies for years", as its content allows scholars to "investigate what distinctions Anglo-Saxons felt important enough to make in the lexicon" (1998). This historical thesaurus, then, is considered a valuable asset to many scholars. Opening up

[1] These numbers are based on an export of the TOE database provided on 26 May 2017.

the knowledge contained within – by providing the thesaurus in an appropriate form – is therefore an important aspect for its use in research.

Work on TOE continued after its first publication in 1995, resulting in further editions. None of these, however, was published in a linguistic linked data form. The benefits promised by such a form – e.g., interoperability and reuse – warrants looking into how such a lexicographic resource can be represented using the relevant standards. This paper therefore details the process of bringing TOE to the Semantic Web. This process, which converts the contents of the current TOE database into the desired linked data form is illustrated with *frēols* (in the sense of 'free, not enslaved', see DOE, s.v. 'frēols adj.') that is positioned in the TOE category "Freedom, being free". This lexical sense and the category it belongs to are depicted in Figure 1 along with relevant context in the form of synonymous senses (cf. *frēot*) and superordinate categories from the topical system.

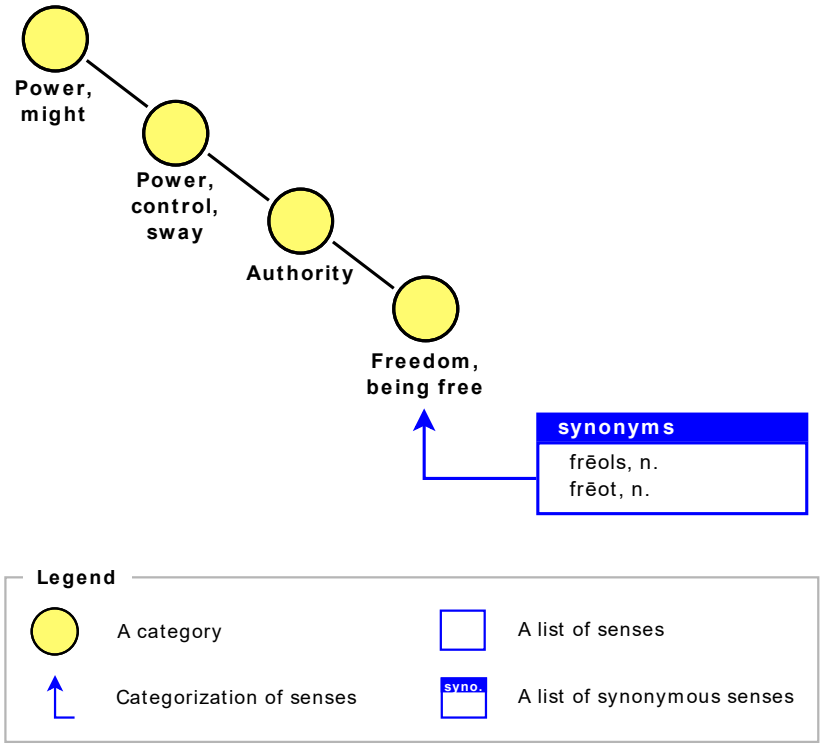

Figure 1: Sample of content from TOE.

In order to discuss the conversion process, we will first continue to describe the current digital form of the TOE database, referred to as its legacy form. The subsequent section provides a better insight into the desired, linguistic linked data form of TOE, which leverages the compact *lemon-tree* model for topical thesauri (Stolk, 2019) alongside the W3C standards OntoLex-Lemon and SKOS (*OntoLex*; *SKOS*). Finally, the conversion

process itself between these two forms is described, followed by the conclusion.

3. Legacy Form

The electronic edition of TOE hosted by the University of Glasgow employs a MySQL database to retrieve and display the thesaurus contents in webpages (TOE, 'Creation of the *Thesaurus*'). The database format is a tabular one, which makes exports possible to other formats that can capture rows and columns (MySQL 5.7 Reference Manual, 'What is MySQL?'). Such formats include Excel spreadsheets and CSV files (MySQL 5.7 Reference Manual, 'Alternative Storage Engines'). In fact, the University of Glasgow provides licensees of the TOE database with a copy by means of such formats. The version of the database provided for this research dates from 26 May 2017.

The TOE database consists of three tables. Each of the tables start with a single row containing the column headings. The rows below it – also known as records – capture instances. The first table discussed here is the category table of TOE, of which the structure is illustrated by Table 1.

catid	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**	**subcat**	**pos**	**heading**	**notes**
1	1								N	Earth, world	
2	1							1	N	As God's creation	xr Religion
3	1							1.01	N	In the beginning	
...	...	...	...	...	...	...	...	...	...	...	...
17187	12	1	1	9				18	V	To accept as a slave	
17188	12	1	1	9				19	V	To bring into bondage	
17189	12	1	1	10					N	Freedom, being free	
17190	12	1	1	10				1	N	Citizenship	
17191	12	1	1	10				2	N	A free man	
17192	12	1	1	10				3	N	A free woman	
17193	12	1	1	10				4	N	Freeman of lowest class	

Table 1: Structure of the TOE category table
(the category "Freedom, being free" is highlighted).

The category table of TOE is used to capture information on categories, where each record represents a single category. The table contains twelve columns in total:

- `catid`: This column acts as primary key, which "uniquely identifies each record in a database table" (*W3Schools.com*, 'SQL Primary Key').

- `t1` to `t7`: These columns capture the location in the taxonomy. Values in `t1` specify the position of the first main category compared to others at the same level, values in `t2` of the second tree level, and so on.

- `subcat`: This column indicates the location further down the taxonomy on a subcategory level (where applicable). Subcategories are distinguished from main TOE categories, which are indicated by `t1` through `t7`, in order to indicate a conceptual level in the taxonomy with smaller semantic differences than is the

case with main categories (TOE, 'Classification'). The subcategory position is not stored separately per subordination step, as the case with `t1` to `t7`, but as a single concatenated string delimited by stops.

- `pos`: This column stores the part of speech associated with a category. An indicated part of speech applies to all lexemes and their senses that are positioned directly at the category (i.e., they are not assigned to subordinate categories). Such a group of lexemes and senses in TOE always shares a single part of speech. Possible values are "aj" for adjective, "av" for adverb, "cj" for conjunction, "in" for interjection, "n" for noun, "p" for preposition, "ph" for phrase, "pn" for pronoun, "v" for verb, "vi" for intransitive verb, and "vt" for transitive verb (which may be monotransitive or ditransitive).

- `heading`: This column contains the name of each category in present-day English.

- `notes`: This column contains notes that are mostly editorial in nature. These include adjustments that have taken effect, matters still to be discussed, and so on. Due to their nature, the notes have so far been left unpublished in both paper and electronic editions.

Table 1 is identified by the key value 17189, called "Freedom, being free", expressed by nouns, and located in the taxonomy at position 12.01.01.10 – the 12th top category, followed by the 1st subordinate one, etc. Note that subordination relations applicable to given categories are not captured explicitly in this table but need to be deduced from the position in the taxonomy. Thus, the "Freedom, being free" category is understood to have the category located at 12.01.01 in the taxonomy as its direct superordinate category: "Authority" (`catid` 169410).

The TOE table discussed next is the category-xref table, of which a sample is shown in Table 2.

xid	catid	refid	tnum
1	18	588	01.03.01.05.01
2	18	9166	05.10.05.04.09
3	45	478	01.02.01.01.03
...	...	...	...
839	17189	16858	11.12.01
840	17189	18102	12.07.03

Table 2: Structure of TOE category-xref table
(the cross-references available at category "Freedom, being free" are highlighted).

Each record in the category-xref table represents a cross-reference in TOE from one category to another. Such a cross-reference indicates a related category that may be of interest to the user, too, but is found in another branch of the taxonomy. The table for

these cross-references contains four columns in total:

- `xid`: This column acts as primary key.
- `catid`: This column acts as foreign key. Such a key links one table to another by means of a reference to a primary key (*W3Schools.com*, 'SQL Foreign Key'). In this case, the column values refer to the primary key of the TOE category table. The categories indicated here are those at which a cross-reference is made.
- `refid`: This column, too, acts as foreign key to the TOE category table. The categories indicated here are those to which a cross-reference is made.
- `tnum`: The values of this column capture the location in the taxonomy of the category referenced in the `refid` column. (Note that this information is superfluous, as it can already be retrieved from the TOE categories table.)

To illustrate, the category "Freedom, being free" (`catid` 17189) has two cross-references: one to category "Absence of restraint, freedom" (`refid` 16858) and one to "Abstinence/exemption (from)" (`refid` 18102). These two categories referred to are found in another branch of the taxonomy than "Freedom, being free". In other words, there exists no subordinate/superordinate relation between them. Hence, the cross-referencing mechanism is employed to indicate that, nonetheless, these categories have a related topic according to the editors.

lid	catid	prefix	word	catorder	et	notes	oflag	pflag	gflag	qflag
1	1		brytengrundas	1		ChristA 355	Y	Y	N	N
2	1		brytenwangas	2		ChristA 380	Y	Y	N	N
3	1		eormengrund	3		Beo 859	Y	Y	N	N
...	...	...	...	...	...	...	...	...	...	...
39486	17187		hēafod niman	1			N	N	N	N
39487	17188	=	(ge)hæftan	1			N	N	N	N
39488	17189		frēols	1			N	N	N	N
39489	17189		frēot	2			N	N	N	N
39490	17190		burhrǣden	1			Y	N	Y	N
39491	17190		burhscipe	2			N	N	N	N
39492	17191		bonda	1	bond		N	N	N	N
39493	17191		ceorl	2	churl		N	N	N	N

Table 3: Structure of TOE lexeme table
(the lexeme frēols that is found at category "Freedom, being free" is highlighted)

From the data it appears that cross-references in TOE occur between main categories only. No cross-references exist from one subcategory to another, from a main category to a subcategory, or vice versa. Thus, although we find "Freedom, being free" is related to "Absence of restraint freedom", no cross-reference is made at one of its subcategories. It is likely that the editors of TOE deemed using cross-references for subcategories to be too fine-grained to indicate and maintain, and therefore kept such references confined to the main categories of the thesaurus. The third and last table of the TOE

legacy form is the lexeme table, depicted in Table 3.

Each record of the lexeme table represents an Old English lexeme that has been categorized based on one of its senses. The table contains eleven columns:

- `lid`: This column acts as primary key.
- `catid`: This column acts as foreign key to the TOE category table and assigns a lexeme, or rather one of the senses of a lexeme, to the category indicated.
- `prefix`: Values in this column, if filled in, can be "+" or "=". These signs correspond to + and ± in the second edition of the Old English dictionary by Clark Hall (CASD)[2]. Its introduction states the following:

  > Words beginning with *ge-* have been distributed among the letters of the alphabet which follow that prefix, and the sign + has been employed instead of *ge-* in order to make the break in alphabetical continuity as little apparent to the eye as possible. The sign ± has been used where a word occurs both with and without the prefix.

  This information on *ge-* prefixes has been superseded in TOE[3]. The current knowledge on prefix use can be deduced from the values in the `word` column.
- `word`: This column contains the head-form of each Old English lexeme. Optional segments of a word (which can be prefixes like *ge-*) are indicated between parentheses. See, for example, the lexeme with `lid` 39487 in Table 3.
- `catorder`: The values of this column indicate the order in which categorized lexemes are to be displayed that are located at the same category.
- `et`: This column contains etymological notes related to the lexeme. For instance, the Old English *ceorl* (`lid` 39493) developed into *churl* (OED, s.v. 'churl, n.').
- `notes`: This column contains notes. These typically mention how often or where a lexeme is found in the Old English corpus. Thus, the noun *eormengrund* (`lid` 3) is noted to be found on line 859 in the poem *Beowulf*.
- `oflag`: This column represents one of the distribution flags of TOE. When the value "Y" is recorded, the word form of the lexeme in question – not in any one specific sense – is marked as "very infrequent" in the Old English corpus.

[2] Information gained in personal correspondence with prof. Marc Alexander (6 August 2017).

[3] One example of knowledge in the `prefix` column being outdated is found with the lexeme with `lid` 582. The `prefix` column suggests the *ge-* prefix of this lexeme is mandatory (+), but the word column indicates that is no longer considered to be the case: "(ge)mȳþe".

- `pflag`: A distribution flag marking those word forms found only in poetry.
- `gflag`: A distribution flag marking those word forms found only in glosses.
- `qflag`: A flag marking word forms as "highly dubious" (TOE, 'Distribution Flags').

To illustrate, the lexeme *frēols* has a sense categorized as belonging to category 17189, "Freedom, being free" (see `lid` 39488). This lexical sense is meant to be displayed as the first one of this category, with the synonymous sense of *frēot* (`lid` 39489) as the second one. The word-forms of *frēols* are not marked as occurring very infrequently in the Old English corpus, in poetry only, in glosses only, or as questionable.

The lexeme table of TOE is rather inefficient for editorial purposes. Each record provides information for a lexeme (such as its head-form, and the distribution of its word forms) but also for a specific sense of that lexeme (such as its placement in the topical system). In fact, the `lid` value of each record is not unique per lexeme. Instead, it is unique per lexical sense. Information on a lexeme is therefore often recorded multiple times and in multiple locations – in a record for each of its senses. When a structure allows redundancy of information, consistency is more difficult to ensure. Contradictory statements are certainly present in the current dataset[4]. Such defects will not be magically mended by porting TOE to linguistic linked data. What the process will do, however, is make a clearer distinction between lexemes (or lexical entries) and lexical senses, which may improve detection of inconsistencies.

4. Linguistic Linked Data Form

A linguistic linked data form for topical thesauri should reuse standardized terminology in order to be interoperable. OntoLex-Lemon and SKOS are highly suitable to this end for capturing both lexical items and a hierarchy of concepts that represent the topical system of a thesaurus. Content from TOE can thus be published on the Web in a form that is machine-interpretable and understood in a wider community. Figure 2 charts, in a coarse manner, the relation between the content from the TOE sample and the linked data terminology from SKOS and OntoLex-Lemon. The relation `a` in this figure, and throughout this paper, is shorthand for `rdf:type` and can be read as "is a" or "is of type" (*RDF 1.1 Turtle*). As can be seen in Figure 2, a categorized lexeme corresponds with a `LexicalSense` in the ontolex module from OntoLex-Lemon. Similarly, a TOE category corresponds with a `LexicalConcept`. Thus, the Old English words *frēols* and *frēot* have lexical senses that lexicalize the concept "Freedom, being free". Superordination between concepts, such as between "Power, control, sway" and "Power,

[4] The noun *earfoþsīþ*, for instance, has two categorized senses in TOE (`lid` 22631 and 32588). Their registered `pflag` values contradict one another – "Y" and "N" respectively – even though both senses share their word forms and the distribution of these forms.

might", is indicated through the `broader` relation from SKOS. A more thorough list of linked data terminology and corresponding TOE content is available in Table 4. Most of the TOE table elements translate directly to linked data counterparts, although there are a few exceptions. These exceptions, discussed below, are taken into account in the linked data form that is proposed for the content of TOE.

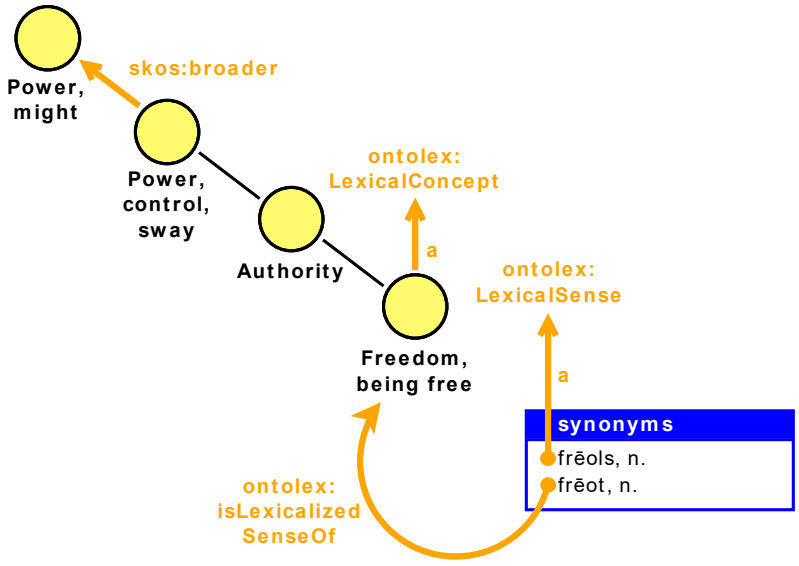

Figure 2: Sample of TOE content and its relation to linked data terminology from OntoLex-Lemon and SKOS

Firstly, some TOE content is not meant to be made available to the general public. Three elements are purely editorial in nature: the `notes` column from the category table and the `et` and `notes` columns from the lexeme table[5]. Various other elements are redundant or have been superseded. These bits may have been useful to the editors during the task of compiling the TOE dataset, but retaining them will likely prove detrimental or confusing. A case in point is the `catorder` column of the lexeme table. Although its values may aid in presenting synonymous senses in the desired order, they do not assist in determining the order for any given selection of senses. As the order of co-ordinate senses in TOE is a largely alphabetical one (with slight adjustments to take into account optional segments, length marks, and symbols specific to Old English), it would be possible – and preferable – to allow visualizations to determine the order of any selection of senses based on their head-forms. To this end, a label intended specifically for machines to order lexemes and their senses according to straightforward string comparison mechanisms (i.e., on ASCII characters only) would be easy to implement and utilize. The `prefix` column from the lexemes table, too, contains

[5] Information gained in personal correspondence with Prof. Jane Roberts (30 August 2017).

information that may best be left unshared with users. Its values are no longer current and can, especially if juxtaposed with the prefix information encoded in the `word` column, confuse users by contradictory statements on whether word forms of a particular lexeme existed with or without the *ge-* prefix. The aforementioned bits of information that are not meant for public consumption should not be part of any publication – including one in a linguistic linked data form.

Secondly, the TOE dataset is in some places more explicit than needed and less explicit in others. The category table, for instance, does not contain a column that explicitly captures the unique id (i.e., a `catid` value) of a superordinate category. As a result, subordination of categories needs to be deduced by means of combining the information from the identification columns – `t1` to `t7` and `subcat` – and comparing the identification values between categories. Storing the identification information separated over various columns hinders both retrieval of the identification string for a category and subsequent comparison of two such strings. Therefore, superordinate categories will be connected explicitly for the linguistic linked data form of TOE. Moreover, the identification string of each category will be stored and offered in a concatenated form rather than broken up in several segments[6].

Thirdly, the TOE dataset conflates information on lexical senses and lexemes into a single structure: the lexeme table. The linked data terminology from OntoLex-Lemon disentangles these two notions, calling the former a `LexicalSense` and the latter a `LexicalEntry`. As the primary key of the lexeme table is unique per sense of a lexeme, each of these records is associated with a `LexicalSense` rather than a `LexicalEntry`. Although the existence and name of a `LexicalEntry` can be deduced from the TOE lexeme table, the TOE dataset contains insufficient information to determine which senses belong to the same lexical entry. According to the specification of OntoLex-Lemon, words "may be different lexical entries if they are distinct in part-of-speech, gender, inflected forms or etymology" (OntoLex). Although TOE indicates the part of speech per lexical sense (i.e., via the `pos` column in the category table), the thesaurus does not currently indicate their gender or inflected forms. As such, a `LexicalEntry` will be created for each `LexicalSense` until information is made available in the future on which of these deduced lexical entries are meant to be one and the same. Such information can be compiled and offered by parties other than the editors of TOE, owing to the new linked data form of the dataset[7].

---

[6] The reason as to why the TOE category table does not store its identification information in a concatenated string but spread over multiple columns is likely found in the development process of the thesaurus, which saw shifts in the technologies used and the identification for categories (TOE, 'Creation of the *Thesaurus*'). One change in the identification system, for instance, is that subcategories have been provided with numbering since the first electronic edition.

[7] Asserting an `owl:sameAs` relation between two `ontolex:LexicalEntry` instances will effectively indicate that the two are to be considered one and the same entry.

Lastly, some of the contents of TOE require linked data terminology that is more specific than that found in SKOS and OntoLex-Lemon alone. To illustrate, a label used to aid computers in determining the presentation order of senses may be a `hiddenLabel` according to SKOS. Such hidden labels are intended for machine processing rather than for people to read. However, the hidden label for TOE should convey that it is specifically meant for the purpose of ordering rather than, for instance, searching alternative spellings. For this label, a new linked data term has been coined for TOE that extends the standardized terminology from SKOS. This coined term can be found in Table 4, including the terminology from SKOS that it extends (indicated through the '>' symbol). Next to this need specific to TOE, two other aspects of this thesaurus are in need of being captured in linked data – aspects shared by a great number of topical thesauri (Stolk, 2019).

The first aspect common in topical thesauri is a division of their topical systems into conceptual levels. As mentioned above, TOE distinguishes two such levels in its database: main categories (simply called categories) and subcategories. The distinction of such levels has been deemed important enough to be included by editors. Indeed, for some thesauri, including TOE, the presentation and navigation mechanisms rely on these distinctions.[8] For a linked data form of TOE, then, this conversion follows the recommendations outlined by the compact *lemon-tree* model, which offers relevant terms such as `ConceptualLevel` and `conceptualDepth` – analogous to how tree levels can be represented using the XKOS (a well-known extension to SKOS used for statistics).

A second aspect, shared by all topical thesauri, is that they categorize lexical items. This is true both for thesauri that group lexical senses into sets of near-synonyms and those that do not. The *lemon-tree* model recognises the need to capture this loose form of categorization, for which it offers the `isSenseInConcept` property and indicates its relation to OntoLex terminology: the *lemon-tree* property is stated to be a more generic form (or super property) of OntoLex `isLexicalizedSenseOf`. This most basic form of categorization found in topical thesauri, then, can be automatically inferred by using the *lemon-tree* model alongside OntoLex for lexical senses in TOE that are asserted to lexicalize a given SKOS `Concept`. Figure 3 illustrates the resulting form for the sample content of TOE used throughout this paper. A combined presentation of this sample content is available in Figure 4. Prefixes are used to abbreviate the namespaces of data vocabularies, for which a mapping is provided in Table 5.

[8] Levels more abstract in nature are typically meant to be navigated first and allow the user to make greater semantic strides, as it were, than conceptual levels more specific in nature.

Linked data property	**Value obtained from legacy form TOE**
ontolex:ConceptSet	
skos:prefLabel	The name of the lexicon as a whole (i.e., "Thesaurus of Old English")
tree:conceptualLevels	An ordered list of the category types distinguished in the lexicon
skos:Collection > tree:ConceptualLevel	
skos:prefLabel	The name of the category type (i.e., "Categories" or "Subcategories")
tree:conceptualDepth	The conceptual depth of the category type
skos:member	The URI for a category belonging to this category type
ontolex:LexicalConcept	
skos:prefLabel	The name of the category
skos:broader	The URI for the superordinate category
skos:notation	The identification of the category
skos:related	The URI for a cross-referenced category
skos:inScheme	The URI for the lexicon as a whole (see ontolex:ConceptSet)
skos:topConceptOf	The URI for the lexicon as a whole (property applicable only to the top-most categories in the lexicon)
ontolex:LexicalEntry	
skos:prefLabel	The name of the lexeme
skos:hiddenLabel > toe:orderLabel	The name of the lexeme, rewritten so as to enable computers to sort these variants alphabetically by conventional means
rdf:type	The URI for the class indicating the part of speech of the lexeme
rdf:type	The URI for the class indicating the distribution of the word forms of the lexeme
ontolex:LexicalSense	
skos:prefLabel	The name of the categorized lexeme
ontolex: isLexicalizedSenseOf	The URI for the category at which the categorized lexeme has been positioned (and is therefore known to lexicalize)
ontolex:isSenseOf	The URI for the ontolex:LexicalEntry associated with the lexeme

Table 4: Linked data terminology and corresponding TOE content
(grey rows across the width of the table state the type of resource that will be formed; subsequent rows indicate which properties will be used to capture information for that resource and what their value will be).

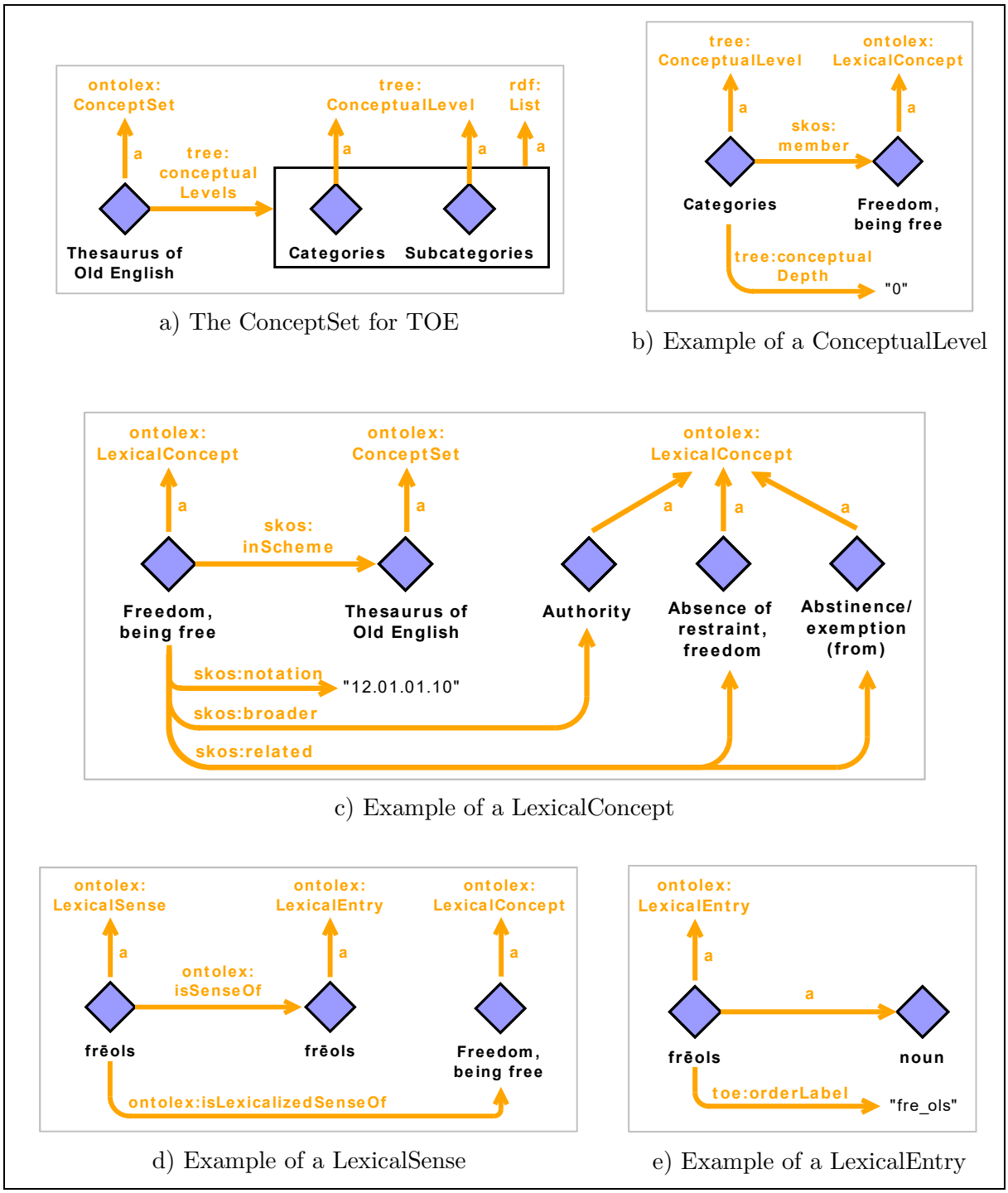

Figure 3: Linguistic linked data form of TOE
(diamonds represent linguistic linked data resources of TOE; arrows represent properties).

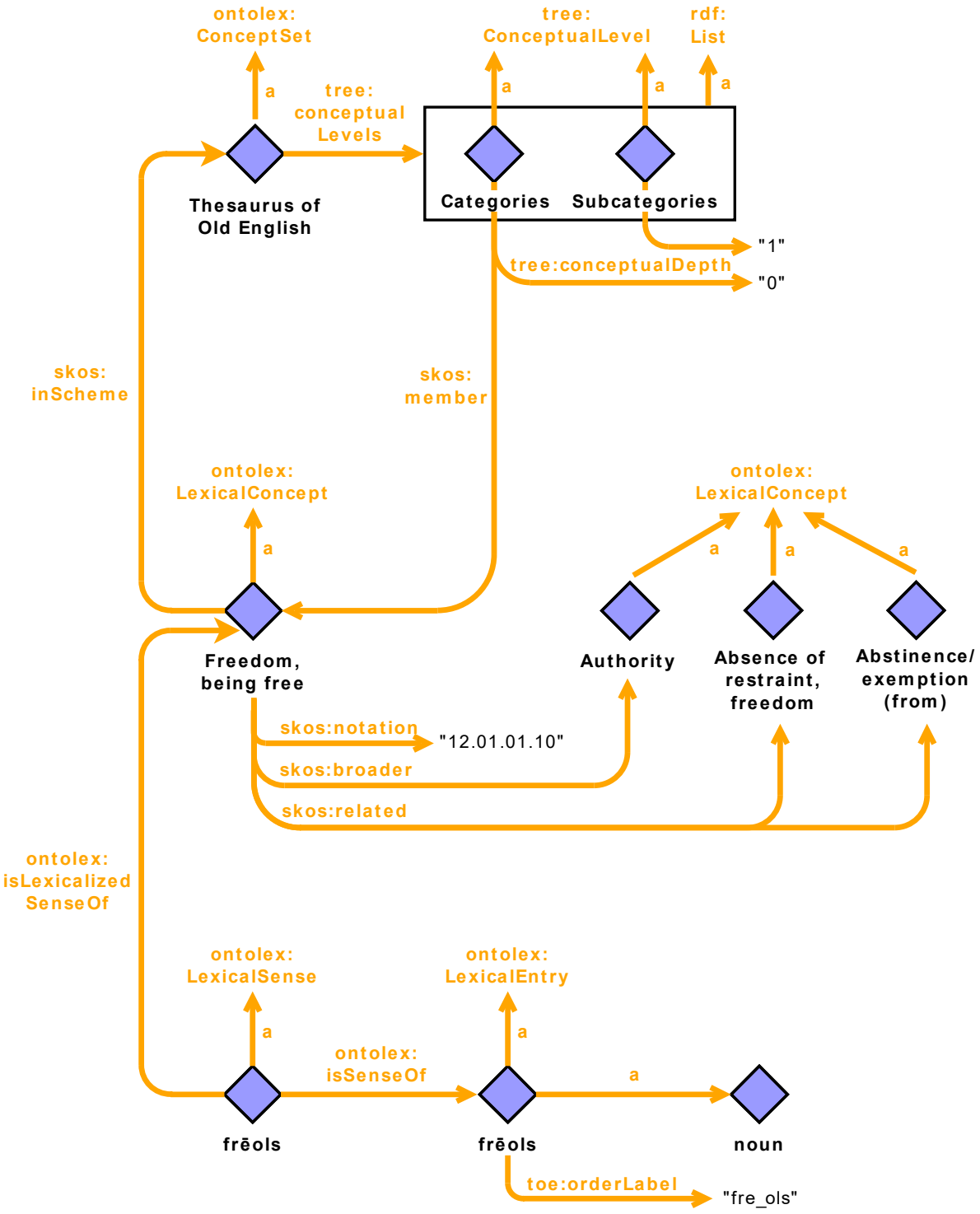

Figure 4: Linguistic linked data form of TOE
(combining the examples provided in Figure 3).

Prefix	Namespace
ontolex:	http://www.w3.org/ns/lemon/ontolex#
owl:	http://www.w3.org/2002/07/owl#
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
skos:	http://www.w3.org/2004/02/skos/core#
toe:	http://oldenglishthesaurus.arts.gla.ac.uk/
tree:	http://w3id.org/lemon-tree#

Table 5: Namespaces.

One further aspect needs to be discussed on bringing TOE content to the Semantic Web: the identification of each resource formed from TOE content. Bits of information on the Semantic Web are identified by a URI, typically in the form of an HTTP address. This holds for terminology from data vocabularies such as SKOS and OntoLex-Lemon, but also for instance data using such terminology. Best practices for coining URIs state that they should be simple, stable, and manageable (CoolURIs; CHIPS; SGOH). The first requirement entails that URIs need to be short and easy to remember; the second that they ought to be independent of the technology used to retrieve or visualize the content (as the software used may change); and the third that issuing new URIs should adhere to a straightforward strategy so as to be able to manage and maintain published content. With these requirements in mind, the following URI strategy has been adopted for the linguistic linked dataset of TOE. Each URI will be formed out of the following segments:

1. the Web domain of TOE (i.e., http://oldenglishthesaurus.arts.gla.ac.uk/),
2. the type of content the URI denotes (e.g., category, sense, entry), and
3. a unique number or string provided by the legacy form, if available.

The TOE category "Freedom, being free" (with `catid` 17189) thus gets the URI http://oldenglishthesaurus.arts.gla.ac.uk/category/#id=17189 for its corresponding `LexicalConcept`. The lexical sense of the lexeme *frēols* (with `lid` 39488) gets http://oldenglishthesaurus.arts.gla.ac.uk/sense/#id=39488. This strategy has an additional advantage: it is aligned with the URI strategy in place for categories in the electronic edition of TOE hosted by the University of Glasgow. As a consequence, one can simply enter the URI of a category in a browser to view human-readable documentation on it. Adding linked data support to the electronic edition of TOE, as hosted by the University of Glasgow, is thus possible in the future without demanding a review or rework of the existing presentation. Having discussed both the original form of the TOE data and the desired linguistic linked data form, this paper will now turn to the conversion method employed to transform the former into the latter.

5. Conversion Process

Free digital tools already exist that facilitate a transformation from data in a tabular format to a linked data form. In selecting appropriate tools for the conversion of TOE from its legacy form to its desired linguistic linked data form, a number of requirements on the process need to be taken into account. These requirements, based on the premise that conversions ought to be reproducible by scholars with minimal effort, are listed in Table 6 and have been categorized according to priority[9]. Two requirements are mandatory, since these ensure an accurate conversion. The first is that the conversion process must accept tabular input either in an Excel spreadsheet or CSV format and provide transformed output in the RDF format (M1). The second requirement is that the process must be able to apply logic that relates the structure of the source to terminology from the desired linked data vocabularies (M2). The conversion logic for the TOE data has been described in Table 4. This logic also demands combining information from multiple tables, available in separate files. To illustrate, most of the information for lexical entries according to OntoLex-Lemon is found in the lexeme table of TOE. The part of speech of such an entry, however, is registered in another table of TOE: the category table.

Next to the requirements that are mandatory, three others have been formulated to which the process should adhere. Although not mandatory for an accurate outcome, these three requirements are geared towards increasing the maintainability and user-friendliness of the process. Firstly, the process should accept conversion logic in a form that has been standardized and is application-independent (S1). The alternative – relying on a format specific to a single tool – would limit the applicability, understandability, and reusability of the captured logic. Considering the availability of specific tooling and continued support from its creators are by no means guaranteed (as indeed seen for a number of conversion tools)[10], great reliance on a single tool should be avoided. Secondly, the process should be executable by scholars without a background in software development (S2). To be more specific, it should be possible to obtain and install the necessary tools without first having to compile the source code. Moreover, the tools should provide a visual user interface rather than only a command-line execution mechanism. Lastly, the conversion process should be automatable so that it can be performed again with minimal effort after an update of the thesaurus data (S3).

The final requirement for the process, assigned a lower priority than the foregoing ones, is meant to facilitate deploying and utilizing the resulting linguistic linked data. Web-based platforms will be able to retrieve and query information from a thesaurus if its

[9] The requirement prioritization follows the MoSCoW principles, developed by Dai Clegg et al. (1994).

[10] Availability and support for the tools AnnoCultor, Aperture, and NOR2O have been discontinued.

conversion output has been stored in a database that facilitates access for linked data technology (C1). A database for linked data content is called a triplestore. Triplestores typically allow accessing their stored content via queries using the standard querying language SPARQL, which web applications can use to interact with the data.

Must haves	
M1	Accept required input and output formats
M2	Apply required logic for conversion
Should haves	
S1	Employ standardized form for logic
S2	Allow for scholars to perform each step
S3	Allow for automation of all steps involved
Could haves	
C1	Store output in a triplestore with a query endpoint

Table 6: Requirements on the conversion process, categorized according to priority

The W3C provides a convenient overview of a number of tools that convert data into RDF (*ConverterToRdf*). Eighteen free tools listed there comply with requirement M1. These tools are listed in Table 7. Five of them appear to be discontinued, that is, they are no longer maintained or offered for download. Nine others do not comply with M2, either because they do not allow applying logic other than their default (Apache Any23) or because they cannot combine information from tables found in separate input files (RDF123; RDF Refine; csv2rdf4lod; Anzo for Excel; TabLinker; Excel2rdf; Sheet2RDF; Spread2RDF). The remaining four tools, then, conform to both mandatory requirements and should be able to convert the TOE legacy form into a linguistic linked data form. These tools are Datalift, Tarql, Virtuoso Sponger, and XLWrap.

One of the four remaining candidate tools for converting TOE data fails to meet requirement S1. This tool, XLWrap, defines its own form for capturing conversion logic, rather than using a standardized form (Langegger, 2017). A number of standardized forms for capturing conversion logic have been recommended by W3C. Two of these are specifically intended for logic converting tabular data into RDF: CSVW and R2RML. Unfortunately, these two forms are unsuitable for the conversion of TOE. The former cannot be used to combine information from multiple input files. The latter facilitates only relational databases as input and cannot be applied to Excel or CSV files. In fact, the three remaining tools – Datalift, Tarql, and Virutoso Sponger – facilitate transformations utilizing another logic form: SPARQL. This query language, standardized by W3C, allows selecting patterns from an RDF source and constructing new RDF data that adheres to desired patterns.

Software	**M1**	**M2**	**S1**	**S2**	**S3**	**C1**
AnnoCultor	*(discontinued)*					
Anzo for Excel	+	-				
Apache Any23	+	-				
Aperture	*(discontinued)*					
Convert2Rdf	*(discontinued)*					
csv2rdf4lod	+	-				
Datalift	+	+	+	+	-	+
Excel2rdf	+	-				
NOR2O	*(discontinued)*					
RDBToOnto	*(discontinued)*					
RDF Refine	+	-				
RDF123	+	-				
Sheet2RDF	+	-				
Spread2RDF	+	-				
TabLinker	+	-				
Tarql	+	+	+	-	+	-
Virtuoso Sponger	+	+	+	-	+	+
XLWrap	+	+	-	-	+	-

Table 7: Software tools and the requirements they meet

The way in which SPARQL is used differs between Tarql on the one hand and Datalift and Virtuoso Sponger on the other. Tarql employs a unique approach by running SPARQL directly on CSV input rather than on RDF data. It does this by emulating patterns have been found based on the tabular input. Datalift and Virtuoso Sponger employ SPARQL in a two-step transformation. First, these tools apply a default, direct mapping to obtain RDF data that is "often more geared towards describing the structure of the data rather than the data itself" (Lefrancois et al, 2017)[11]. This RDF data can subsequently be transformed to RDF data that uses the desired data vocabularies. In this second step, SPARQL (the standard query language for RDF data) is used to select patterns from the RDF source and construct new RDF data that adheres to the desired patterns. Indeed, this two-step approach is one that can be performed by end-users (using tools such as Datalink) but can also be automated (using a direct mapping application and any triplestore that supports SPARQL queries).

[11] The alternative solution proposed by these authors, an extension to SPARQL, appears promising but has not been accepted yet as part of the SPARQL standard proper.

Moreover, this two-step approach is also applicable to formats other than CSV, which may well suit future conversions beyond TOE. The conversion process for TOE, then, will employ the following generic steps:

1. obtain an RDF graph that expresses the structure of the input data
2. store the RDF graph in a triplestore
3. obtain the RDF that adheres to the desired linguistic linked data form through SPARQL queries

Taking these steps will also ensure that the last of the requirements, C1, is met. In other words, the desired linguistic linked data form that has been obtained will be available for queries by platforms that intend to visualize or utilize the thesaurus information. In fact, these three generic steps, here applied to TOE data, should be applicable to the conversion of any topical thesaurus, including those with legacy formats other than tabular data.

For the tabular data of TOE, the first step of the conversion process can be performed by a number of tools. Apache Any23, CSVW implementations[12], Datalift, and Apache Jena all express the structure of such input data in a similar manner. The default logic that these tools share when processing a CSV file is as follows. Firstly, these tools create a node in RDF for each record from the input. Secondly, they add a relation to that node for each of the filled in cell values they encounter. The identification of this relation (i.e., its URI) ends in the column name[13]. An example snippet of such output can be found in Listing 1. To obtain such results using Jena, one simply has to install Apache Jena and run the following command (adjusted to the desired input filename and the output filename):

```
> riot "input.csv" > "output-graph.ttl"
```

[12] See the CSVW report for a list of implementations (*CSVW Reports*).

[13] The initial letter of the column name is capitalized in the case of Apache Any23.

```
_:S39488 <file://C/lexemes.csv#lid> "39488" ;

<file://C/lexemes.csv#catid> "17189" ;

<file://C/lexemes.csv#word> "frēols" ;

<file://C/lexemes.csv#catorder> "1" ;

<file://C/lexemes.csv#oflag> "N" ;

<file://C/lexemes.csv#pflag> "N" ;

<file://C/lexemes.csv#gflag> "N" ;

<file://C/lexemes.csv#qflag> "N" ;

.
```

Listing 1: Snippet of RDF generated in the first step of the conversion process, based on the record for one of the senses of frēols (lid 39488) and expressed in the Turtle syntax.

The second and third steps of the conversion process require a triplestore. For this paper, the RDF4J triplestore is used to illustrate these steps. RDF4J offers a web-based interface, which allows users to set up a new repository for RDF content (see Figure 5) and therein store the intermediate RDF graphs obtained in step 1 (see Figure 6). Each of the graphs is assigned its own context in the repository, which will allow queries in the next step to select content accurately. Table 8 specifies the contexts used in the conversion process.

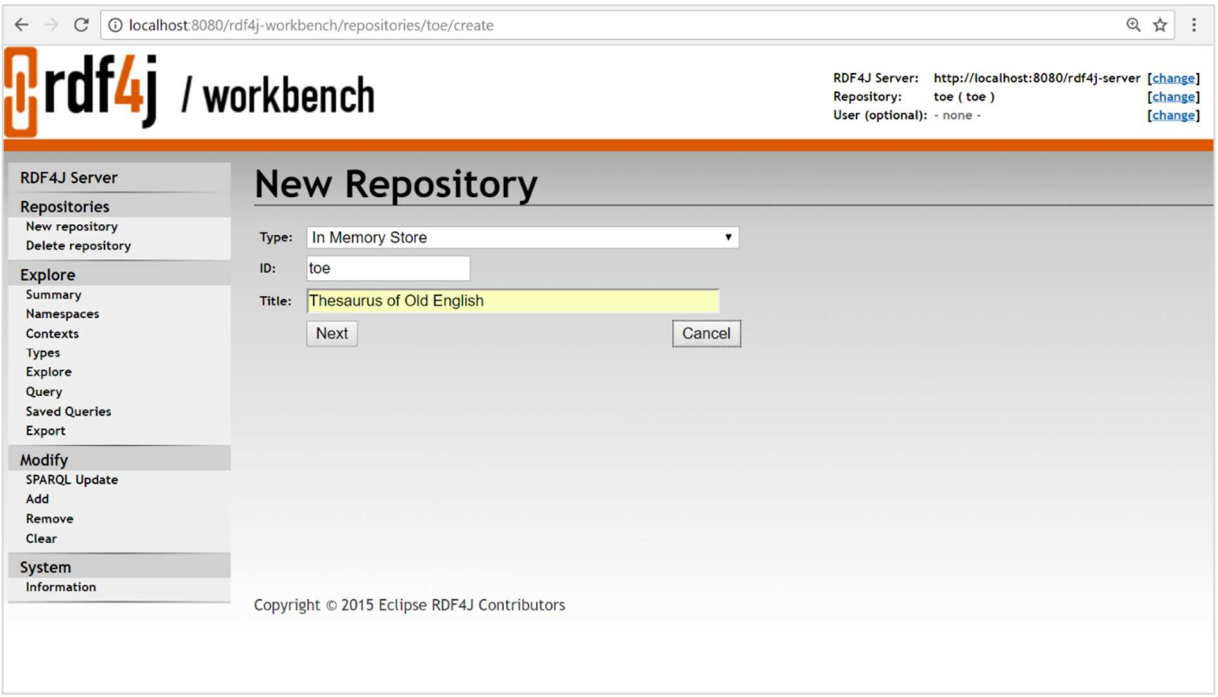

Figure 5: Creating a repository for TOE using the RDF4J user interface

Table of origin	Context
TOE category	<urn:toe:input:category>
TOE category-xref	<urn:toe:input:category-xref>
TOE lexeme	<urn:toe:input:lexeme>

Table 8: Contexts used upon adding RDF to the triplestore.

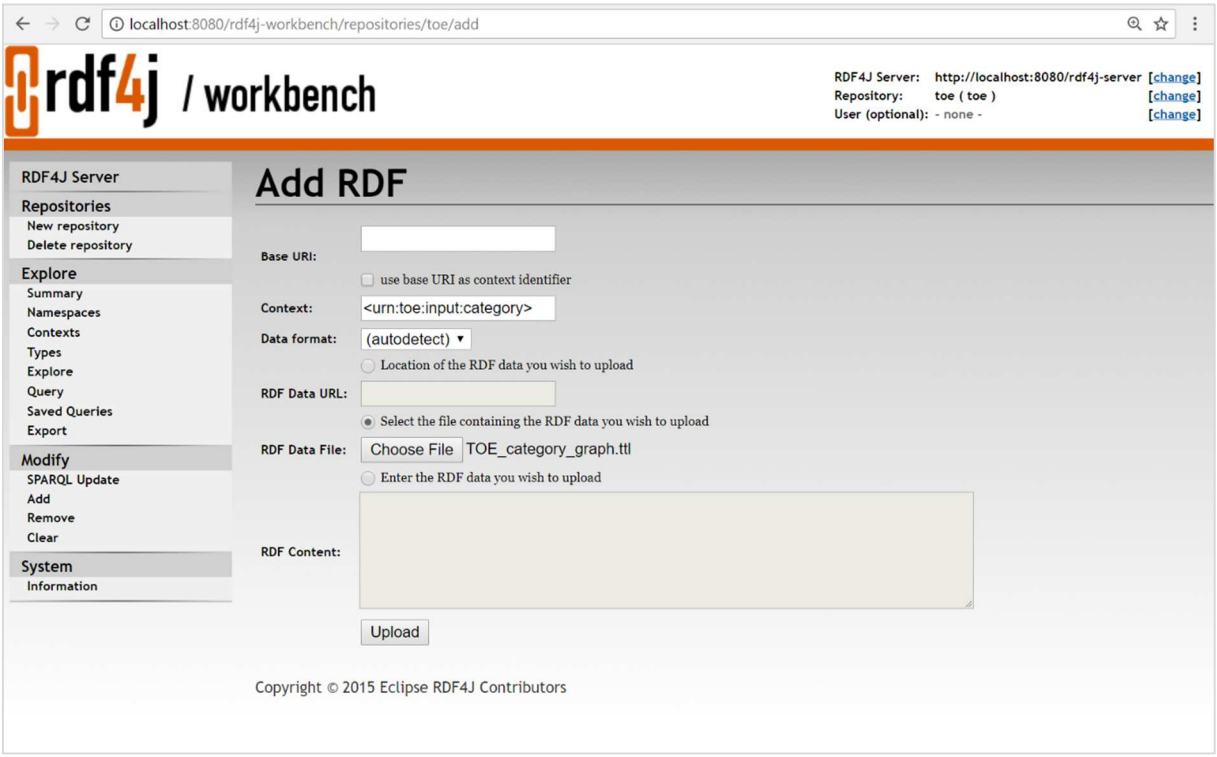

Figure 6: Adding RDF data to TOE categories using the RDF4J user interface.

In the third conversion step, queries are used to transform the available content in the repository to the desired linguistic linked data form. Such queries, written in SPARQL, can be executed via the RDF4J user interface (see Figure 7). Each query specifies a specific pattern that needs to be matched in the available content (in the WHERE clause of the query) and specifies another pattern that should be added as a result for each match (in the INSERT clause). Thus, patterns from the graph content of TOE can be transformed to patterns that conform to the desired outcome.

After the conversion, the resulting RDF will be available for querying and visualization. The intermediate RDF graphs that are uploaded in step 2 can be removed from the triplestore in order to ensure that only the final, desired form of the TOE dataset is indeed available in the repository. Automating the entire conversion process is also possible by means of a batch file. Both the batch file and queries that have been

employed in the conversion of TOE have been made available on GitHub[14].

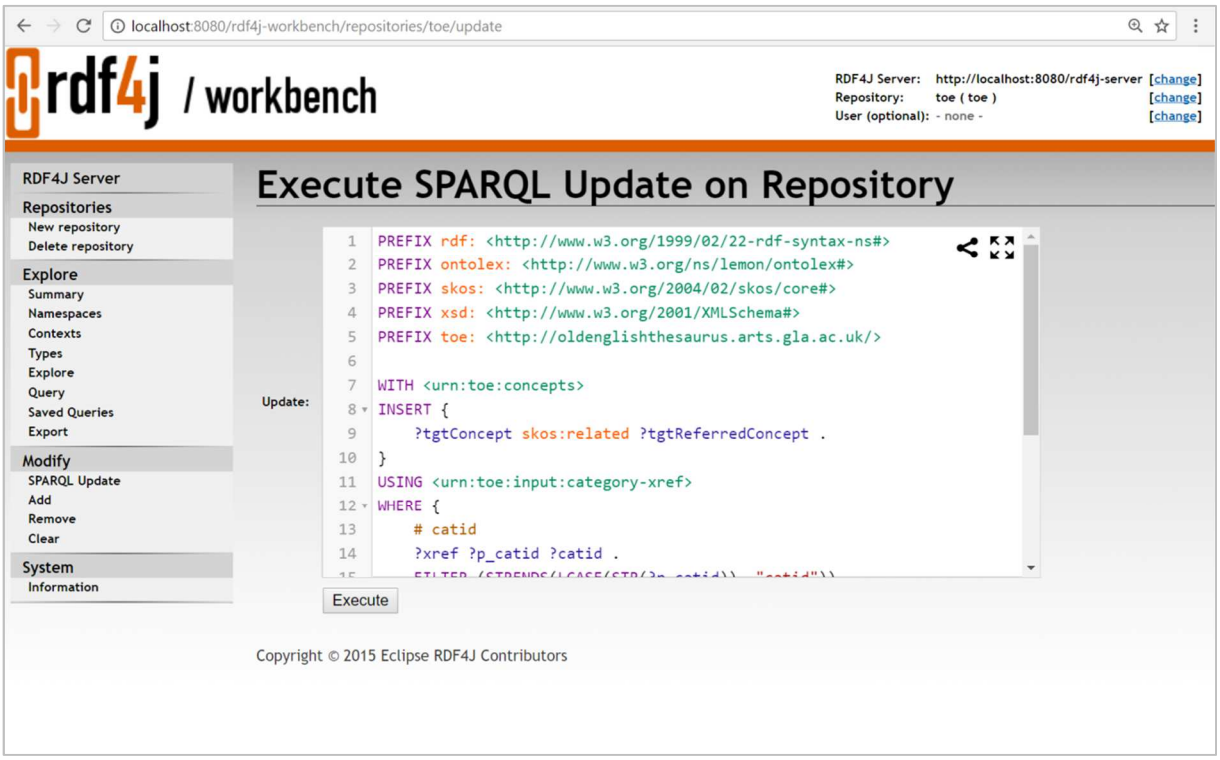

Figure 7: Executing a SPARQL update query using the RDF4J user interface

6. Conclusion

This paper has discussed the conversion of *A Thesaurus of Old English* from its legacy form to a linguistic linked data form utilizing OntoLex-Lemon, SKOS and *lemon-tree*. This conversion follows three steps: 1) obtaining an RDF graph that expresses the structure of the input data, 2) storing the graph in a triplestore, and 3) executing transformation logic using the standardized SPARQL language to produce the desired linguistic linked data form. Using SPARQL for capturing logic rather than a tooling-specific format ensures that the conversion process outlined does not rely on the existence of a single tool. Moreover, the three generic steps of the conversion process should be applicable to the conversion of any topical thesaurus – not just *A Thesaurus of Old English*. The results of the conversion discussed in this paper can be viewed in the online platform Evoke[15].

The new digital form of the thesaurus is used in a number of projects in order to investigate whether linked data mechanisms can facilitate research into Old English language and culture. Some of these projects link lexical items with information to

[14] https://github.com/ssstolk/lld/toe/

[15] http://evoke.ullet.net

indicate their presence in a specific Old English text. Thus, subthesauri can be fashioned to look into specific contexts. Other projects establish links between existing lexicographic resources – connecting ones on Old Dutch and Old Frisian with the thesaurus. Doing so allows for reuse of the thesaurus macrostructure for other languages, but also for contrasting the degree of lexicalization present in these historical languages (e.g., the number of words that we know to have been available in Old Frisian to express a given concept compared to that for Old English). The findings of these and further projects will be presented at the Exploring Anglo-Saxon Eloquence pre-conference workshop at the 21st International Conference of English Historical Linguistics[16].

7. Acknowledgements

The work described in this paper would not have been possible without the support of the Leiden University Centre for Digital Humanities for the Exploring Anglo-Saxon Eloquence project. Special thanks go out to the University of Glasgow, who have been kind enough to provide a license for working with the data of *A Thesaurus of Old English* and to give permission for distributing the resulting linked data form of the thesaurus on the Evoke platform.

8. References

AnnoCultor. Accessed at: https://sourceforge.net/projects/annocultor/. (4 June 2019)

Anzo for Excel. Accessed at: https://supportcenter.cambridgesemantics.com/docs/glossary/Anzo-Excel. (4 June 2019)

Apache Any23. Accessed at: https://any23.apache.org/. (4 June 2019)

Apache Jena. Accessed at: https://jena.apache.org/documentation/io/. (4 June 2019)

Aperture. Accessed at: http://aperture.sourceforge.net/. (4 June 2019)

Bosque-Gil, J. et al. (2016). Modelling Multilingual Lexicographic Resources for the Web of Data: The K Dictionaries Case. In I. Kernerman et al. (eds.) *Proceedings of GLOBALEX'16 workshop at LREC'16*. Portorož, Slovenia, pp. 65–72. Available at: http://www.lrec-conf.org/proceedings/lrec2016/workshops/LREC2016Workshop-GLOBALEX_Proceedings-v2.pdf.

Bremmer Jr, R. H. (2002). Treasure Digging in the Old English Lexicon, Review of *A Thesaurus of Old English*. *NOWELE,* 40, pp. 109–114.

CASD: *A Concise Anglo-Saxon Dictionary for the Use of Students*. (1916). 2nd edition. New York: Macmillan.

CHIPS: *Common HTTP Implementation Problems*. Accessed at: https://www.w3.org/TR/chips/. (9 June 2019)

Clegg, D. & Barker, R. (1994). *Case Method Fast-Track: A RAD Approach*. Boston:

[16] https://icehl21.wordpress.com

Addison-Wesley.

ConverterToRdf. Accessed at: https://www.w3.org/wiki/ConverterToRdf. (20 December 2017)

CoolURIs: *Cool URIs for the Semantic Web*. Accessed at: https://www.w3.org/TR/cooluris/. (9 June 2019)

CSV2RDF: *Generating RDF from Tabular Data on the Web*. Accessed at: http://www.w3.org/TR/csv2rdf/. (9 June 2019)

csv2rdf4lod. Accessed at: https://github.com/timrdf/csv2rdf4lod-automation/wiki. 4 June 2019)

CSVW Reports. Accessed at: https://w3c.github.io/csvw/tests/reports/. (9 June 2019)

Dance, R. (1997). Review of *A Thesaurus of Old English*. *Medium Ævum,* 66(2), pp. 312–313.

Datalift. Accessed at: https://datalift.org/. (4 June 2019)

Declerck, T. et al. (2015). Towards a Pan European Lexicography by Means of Linked (Open) Data. In I. Kosem et al. (eds.) *Proceedings of eLex 2015*. Sussex, United Kingdom, pp. 342–355. Available at: http://www.dfki.de/web/forschung/iwi/publikationen/renameFileForDownload?filename=eLex_2015_22_Declerck+etal.pdf&file_id=uploads_2536.

DOE: *Dictionary of Old English: A to I online*. Accessed at: http://www.doe.utoronto.ca. (4 June 2019)

Evoke. Accessed at: http://evoke.ullet.net

Excel2rdf. Accessed at: https://github.com/waqarini/excel2rdf. (4 June 2019)

Görlach, M. (1998). Review of *A Thesaurus of Old English*. *Anglia* 116(3), pp. 398–401.

Kay, C. & Alexander, M. (2016). Diachronic and Synchronic Thesauruses. *The Oxford Handbook of Lexicography*. Oxford: Oxford University Press, pp. 367–380.

Khan, F. (2016). Representing Polysemy and Diachronic Lexico-semantic Data on the Semantic Web. In *Proceedings of the Second International Workshop on Semantic Web for Scientific Heritage co-located with 13th Extended Semantic Web Conference* (ESWC 2016). Heraklion, Greece, pp. 37–46.

Klimek, B. & Brümmer, M. (2015). Enhancing Lexicography with Semantic Language Databases. In *Kernerman DICTIONARY News,* 23.

Lefrancois, M. et al. (2017). A SPARQL Extension for Generating RDF from Heterogeneous Formats. In *Proceedings of the 14th International Conference of the European Semantic Web Conference*. Portorož, Slovenia, pp. 35–50.

Lemon-tree. Accessed at: https://w3id.org/lemon-tree. (4 June 2019)

MySQL 5.7 Reference Manual. https://dev.mysql.com/doc/refman/5.7/en/. (4 June 2019)

NOR2O. Accessed at: https://github.com/boricles/nor2o. (4 June 2019)

OED: *Oxford English Dictionary Online*. Accessed at: http://oed.com. (4 June 2019)

OntoLex-Lemon*: Lexicon Model for Ontologies*. Accessed at: http://www.w3.org/2016/05/ontolex/. (9 June 2019)

R2RML: *RDB to RDF Mapping Language*. Accessed at:

http://www.w3.org/TR/r2rml/. (9 June 2019)

RDF123. Accessed at: http://ebiquity.umbc.edu/project/html/id/82/RDF123. (20 December 2017

RDF Refine. Accessed at: http://refine.deri.ie/. (20 December 2017)

Roberts, J. (1978). Towards an Old English Thesaurus, *Poetica* 9, pp. 56–72.

Roget: *Thesaurus of English Words and Phrases, Classified and Arranged so as to Facilitate the Expression of Ideas and Assist in Literary Composition*. (1852). London: Longman.

SGOH: *Style Guide for Online Hypertext*. Accessed at: https://www.w3.org/Provider/Style/URI. (4 June 2019)

Sheet2RDF. Accessed at: http://art.uniroma2.it/sheet2rdf/. (4 June 2019)

SKOS: *SKOS Simple Knowledge Organization Reference*. Accessed at: http://www.w3.org/TR/skos-reference/. (9 June 2019)

SPARQL*: SPARQL 1.1 Query Language*. Accessed at: http://www.w3.org/TR/sparql11-query/. (9 June 2019)

Spread2RDF. Accessed at: https://github.com/marcelotto/spread2rdf. (4 June 2019)

Spreadsheet-to-RDF Wrapper. Accessed at: http://xlwrap.sourceforge.net/. (4 June 2019)

Stolk, S. (2019). Lemon-tree: Representing Topical Thesauri on the Semantic Web. In M. Eskevich et al. (eds.) *Proceedings of the 2nd Conference on Language, Data and Knowledge (LDK 2019)*. Leipzig, Germany. Accessible at: https://doi.org/10.4230/OASIcs.LDK.2019.16.

TabLinker. Accessed at: https://github.com/Data2Semantics/TabLinker. (4 June 2019)

TOE: *A Thesaurus of Old English*. Accessed at: http://oldenglishthesaurus.arts.gla.ac.uk. (4 June 2019)

Turtle: *RDF 1.1 Turtle*. Accessed at: https://www.w3.org/TR/turtle/. (9 June 2019)

W3Schools.com. Accessed at: https://www.w3schools.com. (9 June 2019)

XKOS: *An SKOS Extension for Representing Statistical Classifications*. Accessed at: http://www.ddialliance.org/Specification/XKOS/1.0/OWL/xkos.html. (9 June 2019)