Collaborative semantic editing of linked data lexica

John McCrae*, **Elena Montiel-Ponsoda**†, **Philipp Cimiano***

*CITEC, Universität Bielefeld, Universitätsstraße, Bielefeld, D-33615.
†Ontology Engineering Group, Universidad Politécnica de Madrid, Boadilla del Monte, Madrid, Spain
{jmccrae,cimiano}@cit-ec.uni-bielefeld.de, emontiel@delicias.dia.fi.upm.es

Abstract

The creation of language resources is a time-consuming process requiring the efforts of many people. The use of resources collaboratively created by non-linguists can potentially ameliorate this situation. However, such resources often contain more errors compared to resources created by experts. For the particular case of lexica, we analyse the case of Wiktionary, a resource created along wiki principles and argue that through the use of a principled lexicon model, namely *lemon*, the resulting data could be better understandable to machines. We then present a platform called *lemon source* that supports the creation of linked lexical data along the *lemon* model. This tool builds on the concept of a semantic wiki to enable collaborative editing of the resources by many users concurrently. In this paper, we describe the model, the tool and present an evaluation of its usability based on a small group of users.

Keywords: Lexicon, Ontology, Collaborative development

1. Introduction

The web is evolving from a space in which mainly documents are published to a space where 'raw' data is published following the Linked Data principles (Berners-Lee, 2009). Following this trend, there has been substantial interest in applying linked data principles to the publication and linking of language resources, leading to the creation of *lexical linked data* (De Melo and Weikum, 2008; Van Assem et al., 2006). A first step in this direction is the publication of existing lexica and dictionaries as linked data. To support this, the *lemon* model (McCrae et al., 2011) has been proposed as a principled format allowing to publish such resources as linked data. As the creation and curation of such lexical resources often lies beyond the capabilities of single individuals, collaborative lexicon creation projects have started to emerge. Wiktionary for example represents a very successful lexicon project which corroborates the feasibility of a collaborative approach to lexicon creation by a community of users. Wiktionary covers 433 languages and has in fact resulted in a resource which is comparable in size and coverage to curated resources such as WordNet.

One crucial problem with Wiktionary is that its display-orientated markup is difficult to exploit by machines for a number of reasons that we will elaborate below. To remedy this, we propose to use the *lemon* model as a principled model to represent lexica as linked data, and thus we solve the deficiencies of the data model underlying Wiktionary. A natural question that follows from this move is whether the paradigm of collaborative lexicon creation and curation realised in Wiktionary can be successfully transferred to lexica formalised using the *lemon* model. This is the issue we are concerned with in this paper. One inherent problem of this desideratum is that the *lemon* model is difficult to display in contrast to the display-orientated mark-up employed by Wiktionary. In order to support the editing of *lemon* lexica by a larger community of non-professional users, the user interface needs to abstract from the underlying data model, at the same time allowing the users to exploit its expressivity. In this paper, we propose a collaborative approach and an accompanying web-based tool to support the creation of linked data lexica based on the *lemon* model. The paper is structured as follows: in section 2.2. we discuss in more detail the Wiktionary project and highlight some of those deficiencies that make machine processing difficult. We then highlight how a model such as *lemon* provides a principled solution to these problems. Finally, we present *lemon source*, a new approach supported by a web-based tool that allows users to create linked data lexica collaboratively. We discuss the design choices of this approach and contrast it to other approaches allowing users to edit RDF data in a user-friendly way.

2. Background

2.1. Related Work

Most natural language processing applications require some background knowledge (e.g. ontologies, lexica, thesauri) or training data in the form of annotations. Creating such lexical resources, ontologies or annotated corpora is, however, a time-consuming process. Thus, there have been proposals for tool support that allows to share the effort of creating annotated corpora among various parties by fostering collaboration. An example is the OLLIE tool (Cunningham et al., 2003), which is part of the GATE toolkit for natural language processing. This tool is a web application that allows multiple users to collaborate on the task of annotating a corpus. Smooth collaboration is ensured by an architecture that synchronises all annotations via a single relation database back-end such that users can directly see the annotations made by others. Web-based approaches have also been exploited for data collection. Draxler (2006) for instance presents an approach where a large corpus was collected over the web from speakers at schools across Germany.

The use of the web as a data collection method has lead to the creation of many resources from non-expert (and often anonymous) users, a process known as 'crowd-sourcing'. A survey of such approaches is given by Munro et al. (2010).

They conclude that crowd-sourcing “enables systematic, large-scale judgement studies that are more affordable and convenient than ... lab-based studies”. A clear concern with eliciting such information from non-expert users is the correctness of the results. However, studies have found that by combining the annotations of multiple non-expert annotators the resulting annotated data is comparable to the data provided by an expert annotator (see (Snow et al., 2008; Hseuh et al., 2009)).

One of the most prominent examples of such crowd-sourcing is the online encyclopedia Wikipedia [1] and its dictionary version Wiktionary. There have already been several attempts to create a structured resource out of Wiktionary (Zesch et al., 2008a; McCrae et al., 2012b) and to apply the data contained within this resource to NLP tasks such as information retrieval (Müller and Gurevych, 2009) and semantic relatedness (Zesch et al., 2008b). As resources created by crowd-sharing are not typically developed for machine processing, it is a priori not clear how useful such crowd-sourced resources are for NLP tasks. A topological comparative study of Wiktionary as a resource was carried out by Meyer et al. (2010). They found that Wiktionary was similar in structure to resources created by experts (in this case GermaNet and Open Thesaurus), but had fewer semantic links per resource than such resources. They also reported that there were many technical issues in Wiktionary concerning broken links and axiom violations, e.g., the indication of a synonymy between two pages in only one direction, which violates the symmetric axiom for synonymy. McCrae et al. (2012b) presented a study in integrating WordNet with Wiktionary, showing that there was only an approximately 25% overlap at the level of lexical entries between both resources. This suggests that combining these resource may thus be very valuable.

2.2. Wiktionary

In this paper, we take Wiktionary as a representative example of a collaboratively edited dictionary. Wiktionary currently consists of 2.8 million entries (380,000 in English alone)[2]. It would be very useful if Wiktionary could be used to support NLP applications. However, the MediaWiki markup only provides some weak semantic information, as the main purpose of this markup is to display the entries in a uniform manner. Therefore, when attempting to automatically process the markup, a number of issues occur:

- **Implicit Semantics**
  - The markup is mainly used for display purposes and linguistic knowledge is ‘hidden’ behind procedures that render certain templates in an appropriate way. However, this linguistic knowledge, being implicit only, is difficult to access and exploit. This is the case of inflectional markup, e.g. plural formation. The markup often also has additional parameters the semantics of which is not well-defined.
- **Lack of Consistency**
  - Markup is not used consistently in the sense that tags, such as those for part-of-speech, are not used in the same meaning and same purpose across languages.
  - There is no proper typing for the markup, such that the same type of markup is used to mark very different linguistic properties of lexical entries, e.g. using the same type of tags to specify that a lexical entry is `archaic` (representing pragmatic knowledge related to the usage of that entry) or that a lexical entry is `uncountable` (a lexico-syntactic property).
- **Lack of expressivity**
  - Senses are often employed in multiple roles in an entry, e.g. in giving definitions and providing translations. However, there is no explicit ID assigned to theses senses. Thus, individual uses of a sense cannot be consolidated. For example, for the entry “cat” a definition of “Any similar animal of the family Felidae, which includes lions, tigers etc.” is given, but a set of translations is given under the definition “member of Felidae” and a set of synonyms are given under the heading “any member of Felidae”. While it is easy for a human to see that these elements are equivalent, it is a non-trivial task for a machine. An example of this is given in Figure 1.
  - Links to other pages do not specify the particular entry or definition that is relevant. For example, the English “bank” links to the German “Bank” but it is not specified whether the translation is the entry with plural “Banken” (which is correct), or the entry with plural “Bänke” (which is erroneous and means “bench”).
- **Technical inconsistencies**
  - There are often small technical errors. For example, in certain places either ISO 639 codes or language names (in English) may be used to indicate the language of a translation.

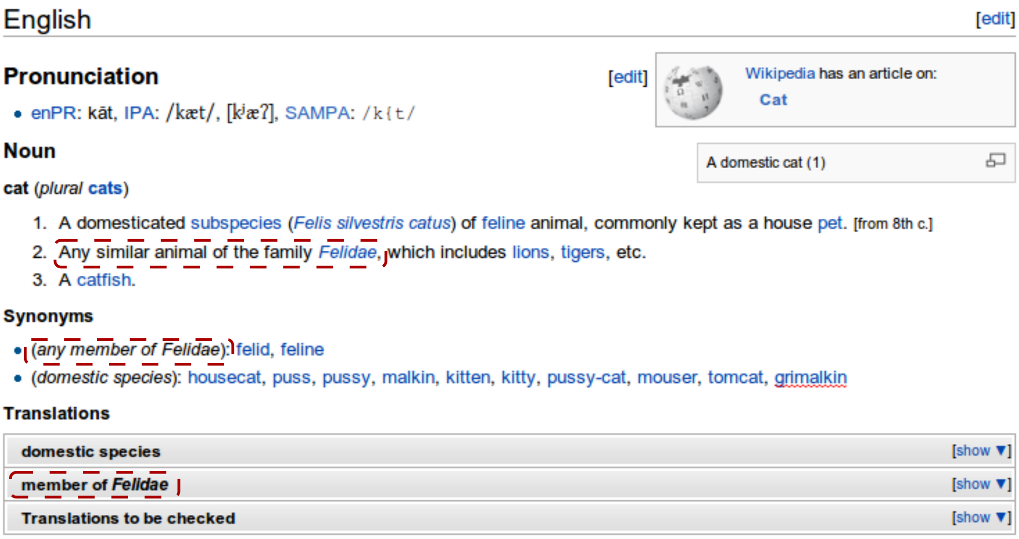
The above issues reveal that we need a sound and linguistically-motivated data model that solves some of these issues, in particular introducing IDs for senses that can be referred to when specifying translations etc. We propose to use the *lemon* model for this, which is briefly described in the next Section.

2.3. The lemon model

The *lemon* model (McCrae et al., 2012a) is a proposed model for the representation of *ontology-lexica* as linked data. The *lemon* model builds on a number of existing standards for the representation of lexica, such as the Lexical Markup Framework (Francopoulo et al., 2006) and the SKOS model (Miles and Bechhofer, 2009) for representing terminologies. With *lemon* we had several key design goals: Firstly, the model is based on RDF, as this is the standard method for distributing linked data on the web.

[1] `http://www.wiktionary.org`

[2] Based on dump dated 08/10/2011



English [edit]

Pronunciation [edit]

- enPR: kăt, IPA: /kæt/, [kʲæʔ], SAMPA: /k{t/

Wikipedia has an article on: **Cat**

A domestic cat (1)

Noun

cat (*plural* **cats**)

1. A domesticated subspecies (*Felis silvestris catus*) of feline animal, commonly kept as a house pet. [from 8th c.]
2. Any similar animal of the family *Felidae*, which includes lions, tigers, etc.
3. A catfish.

Synonyms

- (*any member of Felidae*): felid, feline
- (*domestic species*): housecat, puss, pussy, malkin, kitten, kitty, pussy-cat, mouser, tomcat, grimalkin

Translations

domestic species [show ▼]

member of *Felidae* [show ▼]

Translations to be checked [show ▼]

Figure 1: An example of a Wiktionary entry. The highlighted element indicate semantically related elements that use different markup and labelling. Some content has been removed from this page for illustrative purposes.

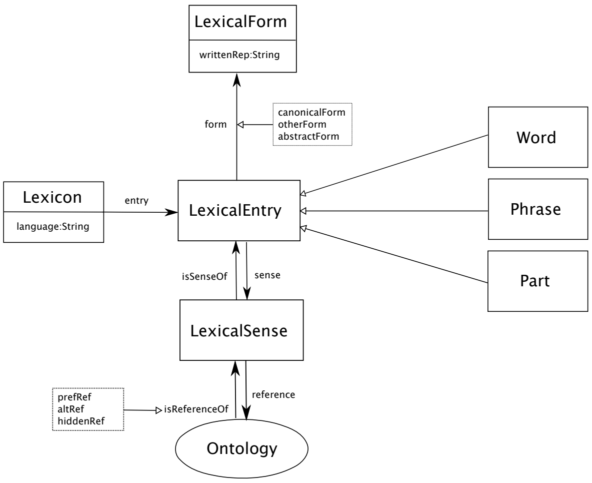


Figure 2: The *lemon* core model

Furthermore, a directed graph model, such as employed by RDF, is more suitable for representing language resources than hierarchical formats (Bird and Liberman, 2001; Ide and Suderman, 2007), e.g., XML. Secondly, the model aims to minimise the number of properties in use so that the model is both concise as well as easy to use and understand. Furthermore, we require that the model is not prescriptive in that it does not make any assumptions about linguistic categories and furthermore supports the use of categories defined in registries such as ISOcat (Kemps-Snijders et al., 2008). Finally, we require that that model uses a process called 'semantics by reference' (Cimiano et al., 2011) in the sense that the semantics of the lexical entries is specified relative to a given ontology (Buitelaar, 2010), which differs significantly from the traditional word sense model employed in most existing lexica, which typically do not define senses by pointing to external concepts.

The *lemon* model consists of a core path illustrated in Figure 2, which consists of the following elements:

- **Lexical Entry**: Representing a single word, phrase or affix ('part').
- **Lexical Form**: Representing inflectional variants of an entry; each form may have a number of different representations in different scripts or phonetic schemes.
- **Lexical Sense**: Reifying the link to the ontology; the lexical sense represents the uses of the entry where it references the concept in the ontology.

In addition, there are a number of modules that build on this core to provide allow to described analysis levels such as phrase structure, morphology, term variation, syntax and argument mapping. More details of these modules and the model in general can be found at `http://lexinfo.net/lemon-cookbook.pdf`.

3. Design

3.1. Motivation

In order to work with lexica in *lemon*, a tool that supports the editing of lexica is needed. While using an existing semantic data wiki for RDF as proposed by Auer et al. (2006), the disadvantage is that it requires that the user has a good command of the data model. If the kind of data that we wish to represent as lexical linked data is highly structured, directly editing the data model may pose difficulties to users that are not well acknowledged with semantic representation languages. To illustrate this, we provide some examples of data model editing (based on the *lemon* model) that would make simple editing impractical.

- **Internal IDs are not meaningful:** In order to represent objects such as senses, forms and (subcategorization) frames in the data model, a reified representation needs to be introduced by assigning a unique ID to the corresponding object. Displaying the associated ID (for example "bank_sense3") would not be meaningful to the user who would clearly prefer a human-readable version of the object.

- **Modelling artifacts require special rendering:** For the representation of certain elements of the model, specialised data structures are required. For example, in the case of RDF, a linked list is required to provide ordered data. Domain-specific data structures (e.g. trees or ordered sequences) need to be rendered intuitively by the editor, but this cannot be done in a generic manner.
- **Consistency of logical units must be maintained:** Some elements in a lexicon should be created and manipulated as a single unit even though they correspond to multiple elements in the data model. For example, subcategorization frames should be created with an appropriate argument structure and given a semantic mapping from the entry's sense.

In addition, as with other lexica models such as LMF (Francopoulo et al., 2006), there is much specialised terminology that is not clear to those without expertise in lexicography or familiarised with the model. For this reason, it is important to provide built-in help that explains and makes accessible the terminology to naive users.

For the above reasons, we thus decided to create a new application from scratch that could provide a clean and intuitive user interface to *lemon* to be used by non-experts to create correct lexica. In addition, we defined the following technical requirements:

- There should be help throughout the system so that the definition of concepts can be provided to non-expert users.
- There should be support for private working spaces for lexica and tracking of changes and status of the lexica.
- The re-use of data categories such as ISOcat (Kemps-Snijders et al., 2008) should be fostered.
- The system should have a model-view-controller architecture, where the model is the RDF store containing the data.
- The data should be accessible by linked data principles. In particular, RDF data should be available by means of "transparent content negotiation" (Holtman and Mutz, 1998) and a SPARQL endpoint should be available.

Our system is designed following the model-view-control pattern where the model is stored in a triple repository. Each modification to the lexicon at the UI is automatically mapped to corresponding changes in the backend, which is implemented by a Virtuoso repository[3]. For the UI we rely on the jQuery library[4], which supports the easy creation of AJAX applications. We employed an extension that allows a help message to be shown on the currently selected element. We also implemented a journalling repository mechanism that allows changes to be tracked and logged to the user that made the modification. User management itself is managed via OpenID, allowing users to use accounts from providers such as Google and Yahoo!. Finally we implemented a linguistic data category ontology interface that uses the LexInfo model[5], which is itself derived from the ISOcat registry (Kemps-Snijders et al., 2008) and is further described in McCrae et al. (McCrae et al., 2011).

3.2. Automatic lexicon creation

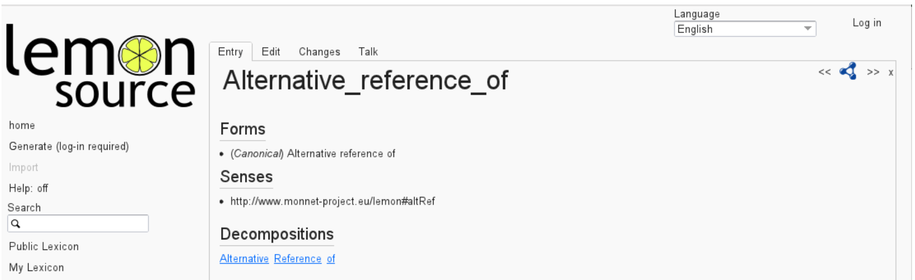
The *lemon source* application supports automatic lexicon creation by means of employing existing NLP systems such as part-of-speech taggers, parsers, tokenizers etc. This system, described in McCrae et al. (2011), allows a lexicon to be created from an input ontology or Linked Data resource automatically. Naturally, the system only allows for automatically generated entries to be added to private users to their set of private lexica, as the result may contain errors and as such requires manual review. This system is implemented by means of a blackboard architecture so that each step of the process is implemented independently, allowing new tasks to be created with ease. Currently, the following processing steps are applied:

- **Label Extraction:** The goal of this pre-processing step is to yield a human-readable label for each ontological element in the ontology or Linked Data resource. Ontologies on the Web and linked data resources differ in how they express lexical information about resources. Some use the `rdfs:label` property, other use `foaf:name` or even other proprietary properties. Thus, specialized procedures are required for label extraction. As ontologies often lack also language information, we also employ a language identification approach and techniques that extract labels from URIs to identify an appropriate label for each resource.
- **Tokenization:** Many labels consist of multiple words that, however, are not separated by blanks. Thus, some special heuristics are needed to tokenize labels.
- **Parsing:** If the label consists of multiple words, we apply a part-of-speech tagger and a parser if there is one available for the language in question. Otherwise, this step is skipped.
- **Tagging:** If a parser is not available, a part-of-speech tagger is applied to infer part-of-speech of the component words.
- **Merging:** The generated entries are compared against the entries in legacy resources such as WordNet or Wiktionary in order to find duplicates. If duplicates are found, the entries are replaced by a URI representing the lexical entry in the legacy resource.

[3] http://virtuoso.openlinksw.com/

[4] http://jquery.com/

[5] http://www.lexinfo.net/ontology/2.0/lexinfo



Alternative_reference_of

Forms

- (*Canonical*) Alternative reference of

Senses

- http://www.monnet-project.eu/lemon#altRef

Decompositions

Alternative Reference of

Figure 3: An example screenshot of *lemon source*

- **Morphological Analysis:** Based on the part-of-speech and canonical form of the entry, a morphological pattern is applied from a pre-loaded set of morphological patterns.
- **Categorization:** In this step a set of specialized rules are applied to the parse tree in order to extract the subcategorization frame of the entry as well as its head.

3.3. Collaborative editing

In this section we present the application we have created to support the creation and collaborative editing of the lexicon associated to ontologies, as well as its publication as linked data. As can be seen in Figure 3, the first step consists in importing an ontology or linked data resource. Then, the application offers the possibility to automatically create a preliminary version of the lexicon based on the natural language information associated with the ontology elements. After this initial step, users can already navigate through the automatically derived lexicon and directly edit it.

As users of the system are intended to be a mixture of people with different degrees of linguistic knowledge, we require that the system provides a number of features for collaborative editing, based on those found in classical data wikis as well as standard practise in the language resource community. As such we formulate the following requirements:

- Support for monitoring changes to a page by means of a change tracker on the RDF model that monitors any changes to an entry and displays them to the user.
- An area for each entry where users can make comments and discuss any details of an entry should there be disagreement.
- Private working spaces for lexica where one or several users may create their own lexica and manage the status of that data before deciding to publish it.
- Statuses for each entry that can be assigned by its editors, such as “Rejected”, “Accepted”, “For review” or “Automatically generated”. These statuses are displayed not only for each entry but also in a summary of all statuses in the lexicon.
- It must be possible to define groups of users who can work on a particular private lexicon and each lexicon should have an owner.
- The owner must be able to publish a lexicon when it has reached a status where it can be published on the Web.

In order to meet these requirements, each entry can be assigned a status from the ones listed below. A manager can then assign roles to the different editors so that they can collaboratively edit the lexica. Once the set of data has reached the ’Accepted’ status, it can be considered ready for publication.

4. Evaluation

For the evaluation of the *lemon source* editor, we performed an evaluation focused on the usability and coverage of the model and the tool. On the one hand, our objective was to find out in how far the *lemon* model is capable of representing the lexical data contained in a resource such as Wiktionary and whether the model matches the requirements that users have for their applications. On the other hand, our purpose is to know users’ opinions on how usable the system is, if the resulting lexicon is as intended, if they easily understand the lexical information captured in the model, if they find it easy to edit, and if the collaborative functionality helps them in creating lexica in an intuitive manner. For these purposes, we conducted an initial set of evaluations with five Masters students, three studying Computer Science, one studying Linguistics and one studying Cognitive Science. They were given a short explanation of the system and allowed to work with the system for about an hour. They were given the task of representing a single entry from Wiktionary for a common term (hence an entry with much information) within *lemon source*. Afterwards, they were asked to answer a questionnaire with ten questions as follows (partially abridged):

1. Did you find the system easy to use?
2. Was the lexical information presented easy to understand?
3. Were you able to represent all information you required?
4. Was the built-in help functionality adequate?
5. Was it straightforward to learn how to use the system?
6. Were the user interface elements clear and understandable?

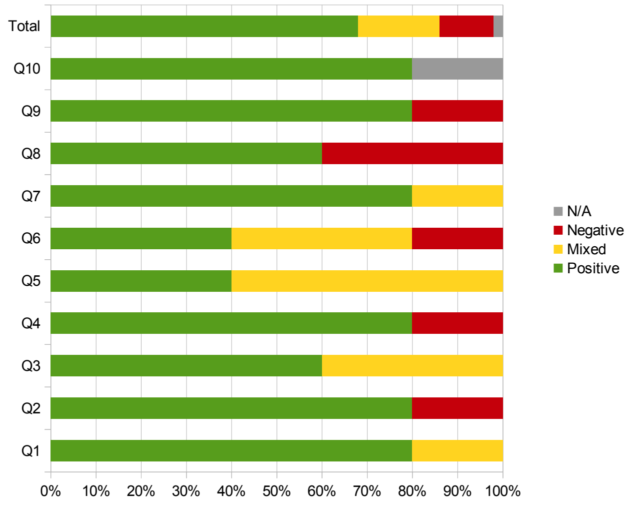


Figure 4: Results of usability study by question

7. Was there too much to read before you could start the system?
8. Was the resulting lexicon as intended?
9. Did straightforward tasks (such as creating a lexical entry with associated subcategorization frame) require too many steps?
10. Did the collaborative functionality help?

The five users were given time to work with the system and then asked to complete the questionnaire. The results are presented in Figure 4 and show that out of a total of 50 responses 34 answers were positive, 6 were negative, 9 were mixed[6] and 1 was not applicable (the user did not use the collaborative tools). In general, the results were positive, and the users were mostly satisfied with the layout of the system, in part due to its similarity to existing Wiki platforms. Most of the negative comments referred to bugs with the system, for example errors in handling strings with apostrophes that are easy to solve. One particular concern that was mentioned (particularly in response to Q2+Q3) was that finding particular linguistic properties or categories was difficult, in particular, one user noted that he/she could not model a verb form as a 'participle', which can be modelled via a property 'verb form mood'. As such we intend to introduce a search function for linguistic properties to enable users to find the correct modelling. One of the CS users found the system to be very difficult to use as he was unfamiliar with most of the linguistic terminology such as 'homonym' and 'phrase tree'. Due to the criticism that the systems lacks documentation, in particular what the description of linguistic categories is concerned, we intend to increase written documentation of the interface and create a video introduction on the main page.

5. Conclusion

We presented a system that is intended for the collaborative creation of linked data lexica using the *lemon* model. This system takes inspiration from the "classical document based wiki" approach but extends this with a structured and linguistically sound data model. We argue that, on the one hand, the data created by classical wikis lacks sufficient semantics to be useful for many text processing applications, and, on the other hand, generic data-driven editors would be difficult to use for non-expert users. Therefore, we argue that for complex language resources, such as lexica, it is necessary to create custom user interfaces that support the creation of high-quality ontology lexica. The system we have presented in this paper, *lemon* source, is a web-based tool that allows users, comprising both experts and non-experts, to collaboratively create an ontology-lexicon semi-automatically based on an automatically created lexicon. An evaluation of the system has shown that the system is usable, but has revealed that appropriate documentation is a key issue that needs to be addressed for the system to be improved.

Acknowledgements

This work was developed in the context of the Monnet project, which is funded by the European Union FP7 program under grant number 248458 and the CITEC excellence initiative funded by the DFG (Deutsche Forschungsgemeinschaft).

6. References

S. Auer, S. Dietzold, and T. Riechert. 2006. OntoWiki–A tool for social, semantic collaboration. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pages 736–749. Springer.

T. Berners-Lee. 2009. Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

S. Bird and M. Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1):23–60.

P. Buitelaar. 2010. Ontology-based Semantic Lexicons: Mapping between Terms and Object Descriptions. In Chu-Ren Huang, Nicoletta Calzolari, Aldo Gangemi, Alessandro Oltramari, Alessandro Lenci, and Laurent Prevot, editors, *Ontology and the Lexicon: A Natural Language Processing Perspective*. Cambridge University Press.

P. Cimiano, P. Buitelaar, J. McCrae, and M. Sintek. 2011. LexInfo: A Declarative Model for the Lexicon-Ontology Interface. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1).

H. Cunningham, V. Tablan, K. Bontcheva, M. Dimitrov, and Ontotext Lab. 2003. Language Engineering Tools for Collaborative Corpus Annotation. In *Proceedings of Corpus Linguistics 2003*, pages 80–87.

G. De Melo and G. Weikum. 2008. Language as a foundation of the semantic web. In *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC)*.

C. Draxler. 2006. Web-based speech data collection and annotation. In *Proceedings of Speech and Computer (SPECOM2006)*, pages 27–34.

[6] The user expressed comments such as "Mostly but..." or expressed negativity about a small part of the system.

G. Francopoulo, M. George, N. Calzolari, M. Monachini, N. Bel, M. Pet, and C. Soria. 2006. Lexical markup framework (LMF). In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'06)*.

K. Holtman and A. Mutz. 1998. Transparent Content Negotiation in HTTP. RFC 2295 (Experimental), March.

P.Y. Hseuh, P. Melville, and V. Sindhwani. 2009. Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 28–35.

N. Ide and K. Suderman. 2007. Graf: A graph-based format for linguistic annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8. Association for Computational Linguistics.

M. Kemps-Snijders, M. Windhouwer, P. Wittenburg, and S.E. Wright. 2008. ISOcat: Corralling data categories in the wild. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'08)*.

J. McCrae, D. Spohr, and P. Cimiano. 2011. Linking lexical resources and ontologies on the semantic web with lemon. *The Semantic Web: Research and Applications*, pages 245–259.

J. McCrae, G. Aguado-de Cea, P. Buitelaar, P. Cimiano, T. Declerck, A. Gómez-Pérez, J. Gracia, L. Hollink, E. Montiel-Ponsoda, D. Spohr, and T. Wunner. 2012a. Interchanging lexical resources on the Semantic Web. *Language Resources and Evaluation*.

J. McCrae, E. Montiel-Ponsoda, and P. Cimiano. 2012b. Integrating WordNet and Wiktionary with lemon. In C. Chiarcos, S. Nordhoff, and S. Hellmann, editors, *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata*, pages 25–34.

C. Meyer and I. Gurevych. 2010. Worth its weight in gold or yet another resourcea comparative study of wiktionary, openthesaurus and germanet. *Computational Linguistics and Intelligent Text Processing*, pages 38–49.

A. Miles and S. Bechhofer. 2009. SKOS Simple Knowledge Organization System reference. W3C Recommendation. Technical report, World Wide Web Consortium.

C. Müller and I. Gurevych. 2009. Using wikipedia and wiktionary in domain-specific information retrieval. *Evaluating Systems for Multilingual and Multimodal Information Access*, pages 219–226.

R. Munro, S. Bethard, V. Kuperman, V.T. Lai, R. Melnick, C. Potts, T. Schnoebelen, and H. Tily. 2010. Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 122–130.

R. Snow, B. O'Conner, D. Jurafsky, and A. Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263.

M. Van Assem, A. Gangemi, and G. Schreiber. 2006. Conversion of WordNet to a standard RDF/OWL representation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC06)*.

T. Zesch, C. Müller, and I. Gurevych. 2008a. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 1646–1652.

T. Zesch, C. Müller, and I. Gurevych. 2008b. Using wiktionary for computing semantic relatedness. In *Proceedings of AAAI*, volume 2008, page 45.