M-ATOLL: A Framework for the lexicalization of ontologies in multiple languages

Sebastian Walter, Christina Unger, and Philipp Cimiano

Semantic Computing Group, CITEC, Bielefeld University

Abstract. Many tasks in which a system needs to mediate between natural language expressions and elements of a vocabulary in an ontology or dataset require knowledge about how the elements of the vocabulary (i.e. classes, properties, and individuals) are expressed in natural language. In a multilingual setting, such knowledge is needed for each of the supported languages. In this paper we present M-ATOLL, a framework for automatically inducing ontology lexica in multiple languages on the basis of a multilingual corpus. The framework exploits a set of language-specific dependency patterns which are formalized as SPARQL queries and run over a parsed corpus. We have instantiated the system for two languages: German and English. We evaluate it in terms of precision, recall and F-measure for English and German by comparing an automatically induced lexicon to manually constructed ontology lexica for DBpedia. In particular, we investigate the contribution of each single dependency pattern and perform an analysis of the impact of different parameters.

1 Introduction

For many applications that need to mediate between natural language and elements of a formal vocabulary as defined by a given ontology or used in a given dataset, knowledge about how elements of the vocabulary are expressed in natural language is needed. This is the case, e.g., for question answering over linked data [23, 20, 10] and natural language generation from ontologies or RDF data [4]. Moreover, in case a system is supposed to handle different languages, this knowledge is needed in multiple languages. Take, for example, the following question from the Question Answering over Linked Data[1] (QALD-4) challenge, provided in seven languages:

1. English: Give me all Australian nonprofit organizations.
2. German: Gib mir alle gemeinnützigen Organisationen in Australien.
3. Spanish: Dame todas las organizaciones benéficas de Australia.
4. Italian: Dammi tutte le organizzazioni australiane non a scopo di lucro.
5. French: Donnes-moi toutes les associations australiennes à but non lucratif.
6. Dutch: Noem alle Australische organisaties zonder winstoogmerk.
7. Romanian: Dă-mi toate organizațiile non-profit din Australia.

[1] www.sc.cit-ec.uni-bielefeld.de/qald/

All these questions can be interpreted as the same, language-independent query to the DBpedia dataset:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT DISTINCT ?uri
WHERE {
       ?uri dbo:type res:Nonprofit_organization .
     { ?uri dbo:locationCountry res:Australia . }
     UNION
     { ?uri dbo:location ?x .
       ?x dbo:country res:Australia . }
}
```

In order to either map the natural language questions to the query or vice versa, a system needs to know how the individual `Nonprofit_organization` is verbalized in the above languages. In addition, it needs to know that the adjective Australian corresponds to the class of individuals that are related to the indivual `Australia` either directly via the property `locationCountry` or indirectly via the properties `location` and `country`. This goes beyond a simple matching of natural language expressions and vocabulary elements, and shows that the conceptual granularity of language often does not coincide with that of a particular dataset.

Such lexical knowledge is crucial for any system that interfaces between natural language and Semantic Web data. A number of models have been proposed to represent such lexical knowledge, realizing what has been called the *ontology-lexicon interface* [18], among them *lemon*[2] [12]. *lemon* is a model for the declarative specification of multilingual, machine-readable lexica in RDF that capture syntactic and semantic aspects of lexical items relative to some ontology. The meaning of a lexical item is given by reference to an ontology element, i.e. a class, property or individual, thereby ensuring a clean separation between the ontological and lexical layer.

We call the task of enriching an ontology with lexical information *ontology lexicalization*. In this paper we propose a semi-automatic approach, M-ATOLL, to ontology lexicalization which induces lexicalizations from a multilingual corpus. In order to find lexicalizations, M-ATOLL exploits a library of patterns that match substructures in dependency trees in a particular language. These patterns are expressed declaratively in SPARQL, so that customizing the system to another language essentially consists in exchanging the pattern library. As input, M-ATOLL takes a RDF dataset as well as a broad coverage corpus in the target language. We present an instantiation of the system using DBpedia as dataset and Wikipedia as corpus, considering English and German as languages to proof that our approach can be adapted to multiple languages. As output, M-ATOLL generates a lexicon in *lemon* format.

The paper is structured as follows: In the following section, we present the architecture of M-ATOLL and discuss its instantiation to both English and German, in particular describing the patterns used for each of these languages. In

[2] `http://lemon-model.net`

Section 3 we evaluate the system by comparing to existing manually constructed lexica for DBpedia. We discuss related work in Section 4, and provide a conclusion as well as an outlook on future work in Section 5.

2 Architecture

In this section we present the architecture behind M-ATOLL. The input is a RDF dataset with or without an underlying ontology as well as a parsed corpus for each of the languages into which the ontology is to be lexicalized; the output is an ontology lexicon in *lemon* format. M-ATOLL comprises two approaches: a *label-based approach* for extracting lexicalizations using ontology labels and additional information, such as synonyms, from external lexical resources, and a *dependency-based approach* for extracting lexicalizations of ontology properties from an available text corpus. We will present both approaches as instantiated for the DBpedia dataset and an English Wikipedia corpus, and then sketch how the system can be ported to other languages, in our case to German.

2.1 Dependency-based approach

Figure 1 presents an overview of the dependency-based approach. The main idea is to start from pairs of entities that are related by a given property, find occurrences of those entities in the text corpus, and generalize over the dependency paths that connect them. The assumption behind this is that a sentence containing both entities also contains a candidate lexicalization of the property in question.

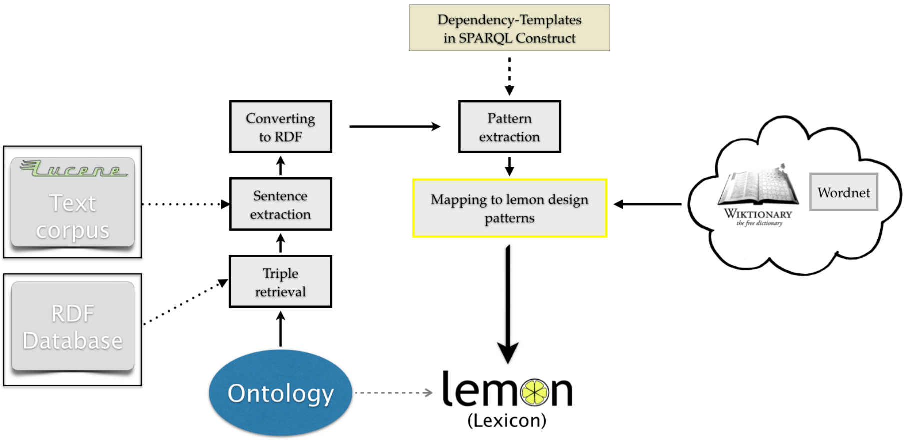

Fig. 1: Overview of the dependency-based approach

First, M-ATOLL expects an index of the available text corpus that stores the dependency parses of all sentences occuring in the corpus in CoNLL format[3]. Using such an index as input instead of raw text increases the flexibility for the adaptation to different languages, as the parsing of a text corpus with a specific dependency parser for the language is an external preparation step. In particular, relying only on an input in CoNLL format keeps the processing itself independent of a specific parser and tag set. In the following we describe all processing steps in detail.

Triple retrieval and sentence extraction Given a property, the first step of M-ATOLL consists in extracting all entities that are connected through the property from a given RDF knowledge base. For the DBpedia property `board`, for example, the following triples are returned (together with 873 other triples):[4]

```
<res:Woolf_Fisher, dbpedia:board, res:Auckland_Racing_Club>
<res:Ram_Shriram,  dbpedia:board, res:Google>
```

For those triples, all sentences that contain both the subject and object labels are retrieved from the text corpus. For example, for the second triple above, one of the retrieved sentences is `Kavitark Ram Shriram is a board member of Google and one of the first investors in Google`. The dependency parse of the sentence is displayed in Figure 2.

Converting parse trees to RDF After extracting all dependency parses of the relevant sentences, they are converted into RDF using our own vocabulary (inspired by the CoNLL format) and stored using Apache Jena[5].

Pattern extraction After storing all parses in an RDF store, dependency patterns that capture candidate lexicalizations are extracted from the parses. In order to minimize noise, we define common, relevant dependency patterns that the extraction should consider. These patterns are implemented as SPARQL queries that can be executed over the RDF store. In this paper we consider the following six dependency patterns (given with an English and a German example each):

1. *Transitive verb*
   - Plato **influenced** Russell.
   - Plato **beeinflusste** Russel.
2. *Intransitive verb with prepositional object*
   - Lincoln **died in** Washington, D.C.
   - Lincold **starb in** Washington, D.C.

[3] `http://nextens.uvt.nl/depparse-wiki/DataFormat`

[4] Throughout the paper, we use the prefix `dbpedia` for `<http://dbpedia.org/ontology/>` and `res` for `<http://dbpedia.org/resource/>`.

[5] `https://jena.apache.org/`

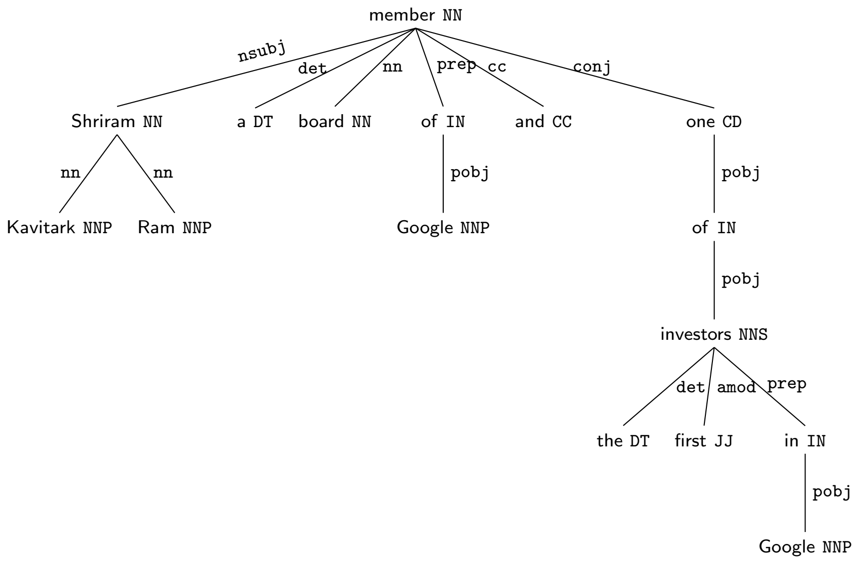

Fig. 2: Dependency tree for the sentence Kavitark Ram Shriram is a board member of Google and one of the first investors in Google

3. *Relational noun with prepositional object (appositive)*
   - Murdoch, **creator of** the Fox Broadcasting Company, retired.
   - Murdoch, der **Gründer der** Fox Broadcasting Company, hat sich zur Ruhe gesetzt.
4. *Relational noun with prepositional object (copulative construction)*
   - Penelope is the **wife of** Odysseus.
   - Penelope is die **Ehefrau von** Odysseus.
5. *Relational adjective*
   - Portugese is **similar to** Spanish.
   - Portugiesisch ist **ähnlich zu** Spanisch.
6. *Relational adjective (verb participle)*
   - Audrey Hepburn was **born in** Belgium.
   - Audrey Hepburn wurde **in** Belgien **geboren**.

Note that these patterns cover relatively general grammatical structures and could be instantiated by several SPARQL queries. The appositive relational noun pattern, for example, is captured by two SPARQL queries that differ only in the direction of a particular dependency relation (both of which occur in the data, leading to the same kind of lexicalizations).

After extracting candidate lexicalizations using these patterns, the final step is to construct a lexical entry. To this end, we use WordNet [14] with the MIT Java Wordnet Interface [6] in order to determine the lemma of a word, e.g marry for the verb form marries, or member for the noun form members. Also, we determine the mapping between syntactic and semantic arguments. For example, in our example in 2, the subject of the property board (Ram_Shriram) corresponds to the subject of the sentence, while the object of the property (Google) corresponds to the prepositional object. The lexical entry created for the noun lexicalization board member then looks as follows:

```
RelationalNoun("board member",dbpedia:board,
  propSubj = CopulativeArg,
  propObj  = PrepositionalObject("of"))
```

This entry makes use of one of the macros for common lexicalization patterns defined in [13], a relational noun macro representing the prototypical syntactic frame x is a board member of y. This entry is equivalent to the following RDF representation:

```
:boardMember a lemon:LexicalEntry;
  lexinfo:partOfSpeech lexinfo:noun;
  lemon:canonicalForm [ lemon:writtenRep "board member"@en ];
  lemon:sense [ lemon:reference dbpedia:board;
                lemon:subjOfProp :x;
                lemon:objOfProp  :y ] ;
  lexinfo:synBehavior [ a lexinfo:NounPPFrame;
                lexinfo:copulativeArg       :x;
                lexinfo:prepositionalObject :y ].

:y lemon:marker [ lemon:canonicalForm
                [ lemon:writtenRep "of"@en ]].
```

For each generated lexical entry we also store how often it was generated and with which SPARQL query it was retrieved.

2.2 Label-based approach

The label-based approach to the induction of lexical entries differs from the dependency-based approach described above in that it does not rely on a text corpus but only on the label of the ontology element in question (classes and properties) as well as on external lexical resources to find possible lexicalizations.

In particular, we use BabelNet [16] for finding synonyms. Currently we pick the first synset that is returned, but we plan to disambiguate the relevant synset by extending our approach to use Babelfy [15], using Wikipedia articles as disambiguation texts.

The label of the DBpedia class Activity, for example, is activity, for which we retrieve the synonym action from BabelNet. The following lexical entries are generated:

```
ClassNoun("activity",dbpedia:Activity)
ClassNoun("action",dbpedia:Activity)
```

The same processing is done for labels of properties, yielding, for example, the following entries for the property spouse:

```
RelationalNoun("spouse",dbpedia:spouse,
  propSubj = PossessiveAdjunct,
  propObj  = CopulativeArg)

RelationalNoun("partner",dbpedia:spouse,
  propSubj = PossessiveAdjunct,
  propObj  = CopulativeArg))

RelationalNoun("better half",dbpedia:spouse,
  propSubj = PossessiveAdjunct,
  propObj  = CopulativeArg)
```

2.3 Adaptation to other languages

This section gives an overview on how to adapt the dependency-based approach of M-ATOLL to other languages, in our case German, for which we present results in Section 3.

The adaptation of the label-based approach largely depends on the availability of external lexical resources, such as BabelNet, for the target language.

In order to adapt the dependency-based approach to German, we first parsed a corpus of around $175,000$ sentences (sentences related to the QALD-3 lexicalization task) from the German Wikipedia, using the ParZu dependency parser [19], storing the resulting parses in CoNLL format in our corpus index. The ParZu parser has the advantage to also lemmatize the tokens of an input sentence, e.g. if the past tense verb form is heiratete, the parser also returns the infinitive verb form heiraten. Therefore no additional resources, such as WordNet, for retrieving the lemma of a word were needed. The next and final step for the adaptation is defining relevant dependency patterns as SPARQL queries in order to retrieve candidate lexicalizations, based on the part-of-speech tag set and dependency relations used by the parser. To this end, we transformed the SPARQL queries used for English into SPARQL queries that we can use for German. This mainly consisted in exchanging the part-of-speech tags and dependencies.

In general, the adaptation of queries to other languages might also involve changing the structure of the dependency patterns it queries for, but the patterns we currently employ are general enough to work well across languages that are structurally similar to English.

3 Evaluation

In this section we describe the evaluation measures and datasets, and then discuss results for English and German.

3.1 Methodology and datasets

For English, we developed M-ATOLL using both training and test data of the ontology lexicalization task of the QALD-3 challenge [5] as development set, i.e. for creating the dependency patterns we query for. It comprises 20 DBpedia classes and 60 DBpedia properties that were randomly selected from different frequency ranges, i.e. including properties with a large amount of instances as well as properties with very few instances. M-ATOLL was then evaluated in terms of precison, recall and F-measure on the manually constructed English *lemon* lexicon for DBpedia[6] [21]. It comprises 1,217 lexicalizations of 326 classes and the 232 most frequent properties. From this dataset we removed all classes and properties used for development, in order to avoid any overlap, leaving a test dataset that is approximately 14 times bigger than the training dataset. As text corpus we use around 60 million parsed sentences from the English Wikipedia.

For German, we use the train/test split of the ontology lexicalization task of the QALD-3 challenge, and evaluate the approach with respect to a preliminary version of a manually constructed German *lemon* lexicon for DBpedia. This results in a training set of 28 properties and a test dataset of 27 properties (all those properties from the QALD-3 dataset that have lexicalizations in the gold standard lexicon). As text corpus, we use around 175,000 parsed sentences from the German Wikipedia.

3.2 Evaluation measures

For each property and class, we evaluate the automatically generated lexical entries by comparing them to the manually created lexical entries in terms of lexical precision, lexical recall and lexical F-measure at the lemma level. To this end, we determine how many of the gold standard entries for a property are generated by our approach (recall), and how many of the automatically generated entries are among the gold standard entries (precision), where two entries count as the same lexicaliztation if their lemma, part of speech and sense coincide. Thus lexical precision P_{lex} and recall R_{lex} for a property p are defined as follows:

$$\begin{aligned} P_{lex}(p) &= \frac{|entries_{auto}(p) \cap entries_{gold}(p)|}{|entries_{auto}(p)|} \\ R_{lex}(p) &= \frac{|entries_{auto}(p) \cap entries_{gold}(p)|}{|entries_{gold}(p)|} \end{aligned}$$

Where $entries_{auto}(p)$ is the set of entries for the property p in the automatically constructed lexicon, while $entries_{gold}(p)$ is the set of entries for the property p in the manually constructed gold lexicon. The F-measure $F_{lex}(p)$ is then defined as the harmonic mean of $P_{lex}(p)$ and $R_{lex}(p)$, as usual.

[6] https://github.com/cunger/lemon.dbpedia

All measures are computed for each property and then averaged for all properties. In the sections below, we will report only the average values.

As mentioned in Section 2, for each generated lexical entry we store how often it was generated. This frequency is now used to calculate a probability expressing how likely it is that this entry is used to lexicalize a particular property in question.

3.3 Results for English

Figure 3 shows results of the dependency-based approach on the training and test dataset in terms of precision, recall and F-measure, considering the top-k generated lexical entries, with $k = 1, 5, 10, 15, 20$ as well as considering all generated entries. The best precision (0.47 on train and 0.44 on test) is reached with $k = 1$, while the best recall (0.29 on train and 0.32 on test) is reached when considering all candidate entries, which also yields the best F-measure (0.30 on train and 0.35 on test).

	Training			Test		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Top 1	**0.47**	0.11	0.18	**0.44**	0.06	0.11
Top 5	0.37	0.20	0.26	0.42	0.19	0.26
Top 10	0.33	0.22	0.27	0.40	0.24	0.30
Top 15	0.32	0.24	0.27	0.40	0.27	0.32
Top 20	0.31	0.26	0.28	0.39	0.27	0.31
All	0.30	**0.29**	**0.30**	0.37	**0.32**	**0.35**

Fig. 3: Results of the dependency-based approach on the English dataset

Figure 4 presents the overall results on the English training and test set for the label-based approach, the dependency-based approach, and when combining both approaches. For performance reasons (especially for the test dataset) we limited the number of considered entity pairs per property to 2,500 pairs, although taking more entity pairs into account will increase the recall significantly, as preliminary tests showed.

Note that the label-based and the dependency-based approach complement each other in the sense that they find different lexicalizations, which leads to an increased recall when combining both.

Finally, Figure 5 shows the contribution of each dependency pattern for English to the results over the training and test sets, when taking all generated entries into account.

3.4 Results for German

Figure 6 shows the results of the dependency-based approach on the training and test dataset for German. As for English, the highest precision is reached with

	Training			Test		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Dependency-based	0.30	0.29	0.30	0.37	0.32	0.35
Label-based	**0.53**	0.24	0.33	**0.56**	0.30	0.40
Both	0.35	**0.44**	**0.39**	0.43	**0.43**	**0.43**

Fig. 4: Overall results on the English dataset, considering all generated entries

	Training			Test		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Transitive Verb	0.48	0.06	0.10	0.47	0.07	0.13
Intransitive Verb with prepositional object	0.41	0.12	0.18	0.43	0.10	0.16
Relational noun (appositive) with prepositional object	0.42	0.04	0.07	0.44	0.07	0.12
Relational noun (copulative) with prepositional object	0.42	0.04	0.07	0.46	0.07	0.12
Relational adjective	0.79	0.04	0.07	0.70	0.02	0.04
Relational adjective (verb participle)	0.40	0.08	0.13	0.45	0.06	0.10

Fig. 5: Contribution of each dependency pattern for English to the results over training and test, taking all generated entries into account

the lowest k, while the highest recall and F-measure are achieved with higher k or considering all candidate entries.

	Training			Test		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Top 1	**0.57**	0.02	0.04	**0.63**	0.06	0.11
Top 5	0.56	0.04	0.07	0.52	0.07	0.13
Top 10	0.56	0.04	0.07	0.50	**0.08**	**0.15**
Top 15	0.55	0.06	0.10	0.50	**0.08**	**0.15**
Top 20	0.55	**0.07**	0.12	0.50	**0.08**	**0.15**
All	0.55	**0.07**	**0.13**	0.50	**0.08**	**0.15**

Fig. 6: Results of the dependency-based approach on the German dataset

The main reason for recall being so low is the rather small set of sentences contained in the text corpus sample. As a result, the approach finds candidate lexicalizations for a bit less than half of the properties. A manual inspection of the generated lexicalizations shows that the found candidates are indeed appropriate. For example, for the property spouse, our approach finds the lexicalizations heiraten (English to marry), Ehefrau von (English wife of), Gatte von (English husband of), leben mit (English to live with), among others, and for the property source (specifying the source of a river), our approach finds the lexicalizations entspringen in (English to originate from) and beginnen (English to begin), among others. We are therefore optimistic that moving to a larger corpus for German will yield results similar to the ones achieved for English.

3.5 Discussion of results

Overall, the English results of the joining the label-based and the dependency-based approach over the test dataset is decent but still far from being able to be used in a fully automatic setting. Roughly, every second lexical entry that is generated is not appropriate. In fact, we rather envision our approach as the basis of a semi-automatic scenario, in which lexical entries are generated automatically and then are manually checked by a lexicon engineer and corrected if necessary. From this perspective, our approach has a clear potential to reduce the amount of manual work required to develop a high-quality lexicon.

The current lack in recall for English is mainly due to the limited number of defined dependency patterns. In addition, not all relevant lexicalizations occur in the available corpus. For example, for the property birthDate the gold standard lexicon contains three entries: born, birth date and date of birth, but only the first one occurs in the Wikipedia corpus in combination with one of our entity pairs from the given property. Capturing a wider variety of lexicalizations thus would require moving to a larger scale (and more diverse) corpus.

Also note that our approach only extracts verbalizations of classes and properties, not of property chains (e.g. grandchild as verbalization of child ∘ cild) or property-object pairs (e.g. Australian as verbalization of country Australia). Dealing with conceptual mismatches between ontological structures and natural language verbalizations will be subject of future work.

4 Related work

An approach to extracting lexicalization patterns from corpora that is similar in spirit to our approach is *Wanderlust* [2], which relies on a dependency parser to find grammatical patterns in a given corpus—Wikipedia in their case as in ours. These patterns are generic and non-lexical and can be used to extract any semantic relation. However, *Wanderlust* also differs from our approach in one major aspect. We start from a given property and use instance data to find all different lexical variants of expressing one and the same property, while *Wanderlust* maps each dependency path to a different property (modulo some postprocessing to detect subrelations). They are therefore not able to find different variants of expressing one and the same property, thus not allowing for semantic normalization across lexicalization patterns.

Another related tool is DIRT [9] (*Discovery of Inference Rules from Text*), also very similar to *Snowball* [1], which is based on an unsupervised method for finding inferences in text, thereby for example establishing that x is author of y is a paraphrase of x wrote y. DIRT relies on a similarity-based approach to group dependency paths, where two paths count as similar if they show a high degree of overlap in the nouns that appear at the argument positions of the paths. Such a similarity-based grouping of dependency paths could also be integrated into our approach, in order to find further paraphrases. The main difference to our approach is that DIRT does not rely on an existing knowledge base of instantiated triples to bootstrap the acquisition of patterns from textual data, thus being completely unsupervised. Given the fact that nowadays there are large knowledge bases such as Freebase and DBpedia, there is no reason why an approach should not exploit the available instances of a property or class to bootstrap the acquisition process.

A system that does rely on existing triples from a knowledge base, in particular DBpedia, is BOA [7]. BOA applies a recursive procedure, starting with extracting triples from linked data, then extracting natural language patterns from sentences and inserting this patterns as RDF data back into the Linked Data Cloud. The main difference to our approach is that BOA relies on simple string-based generalization techniques to find lexicalization patterns. This makes it difficult, for example, to discard optional modifiers and thus can generate a high amount of noise, which has been corroborated by initial experiments in our lab on inducing patterns from the string context between two entities.

Espresso [17] employs a minimally supervised bootstrapping algorithm which, based on only a few seed instances of a relation, learns patterns that can be used to extract more instances. *Espresso* is thus comparable to our approach in the sense that both rely on a set of seed sentences to induce patterns. In our case, these are derived from a knowledge base, while in the case of *Espresso* they are manually annotated. Besides a constrast in the overall task (relation extraction in the case of *Espresso* and ontology lexicalization in our case), one difference is that *Espresso* uses string-based patterns, while we rely on dependency paths, which constitutes a more principled approach to discarding modifiers and yielding more general patterns. A system that is similar to *Espresso* and uses dependency paths was proposed by Ittoo and Bouma [8]. A further difference is that *Espresso* leverages the web to find further occurrences of the seed instances. The corpus we use, Wikipedia, is bigger than the compared text corpora used in the evaluation by *Espresso*. But it would be bery interesting to extend our approach to work with web data in order to overcome data sparseness, e.g. as in [3], in case there is not enough instance data or there are not enough seed sentences available in a given corpus to bootstrap the pattern acquisition process.

The more recent approach by Mahenda et al. [11] also extracts lexicalizations of DBpedia properties on the basis of a Wikipedia corpus. In contrast to our approach, they do not consider the parse of a selected sentence, but the longest common substring between domain and range of the given property, normalizing it by means of DBpedia class labels, such as Person or Date.

Another multilingual system is *WRPA* [22], which extracts English and Spanish lexicalization patterns from the English and Spanish Wikipedia, respectively. Like other approaches, WRPA, considers only the textual pattern between two anchor texts from Wikipedia, no parse structure. WRPA is applied to four relations (date of birth, date of death, place of birth and authorship) on an English and Spanish corpus.

5 Conclusion and future work

We presented M-ATOLL as a first approach for the automatic lexicalization of ontologies in multiple languages and instantiated it for DBpedia in English and German. It employs a combination of a dependency-based and a label-based approach, benefiting from the complementary lexicalizations they find. Furthermore, by extracting candidate lexicalizations by means of matching dependency parses with pre-defined dependency patterns, implemented as SPARQL queries, M-ATOLL offers much flexibility when adapting it to other languages.

However, M-ATOLL is still limited to a few dependency patterns, capturing the most basic grammatical structures. One main goal for future work thus is to increase recall by including more specialized structures. In order to minimize the manual effort in doing so, we intend to develop a procedure for automatically generating relevant patterns along the following lines: On the basis of already existing entries (either extracted by means of some general pre-defined patterns, or part of a gold standard lexicon), we will automatically generate SPARQL queries that retrieve the necessary parts from all sentences that contain the entity labels and the canonical form of the lexical entry. In a next step, these SPARQL queries will be generalized into commonly occurring patterns. This method can also reduce the cost of adapting M-ATOLL to other languages, as the process only needs a few basic patterns in order to bootstrap the pattern learning step would then provide the basis for the large-scale extraction of lexical entries.

Furthermore, we plan to improve the ranking of the generated lexical entries. Currently only the frequency of a certain entry is taken into account, whereas also the frequency of the underlying entity pair plays a role. For example, for properties that have an overlap in their entity pairs, the same verbalizations would be found. Confusing these lexicalizations could be avoided by ranking entries lower that were generated on the basis of entity pairs that also occur with other properties.

Moreover, we want to extend the evaluation of the German ontology lexicalization, running it on a much larger corpus. We plan to instantiate the approach also for Spanish, and finally intend to show that it can be adapted easily not only to other languages but also to other ontologies.

Acknowledgment This work has been funded by the European Union's Seventh Framework Programme (FP7-ICT-2011-SME-DCL) under grant agreement number 296170 (PortDial).

References

1. Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.
2. Alan Akbik and Jürgen Broß. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of the Workshop on Semantic Search in Conjunction with the 18th Int. World Wide Web Conference*, 2009.
3. Sebastian Blohm and Philipp Cimiano. Using the web to reduce data sparseness in pattern-based information extraction. In *Knowledge Discovery in Databases: PKDD 2007*, pages 18–29. Springer, 2007.
4. Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. Natural language generation in the context of the semantic web. *Semantic Web*, 2013.
5. Philipp Cimiano, Vanessa Lopez, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. Multilingual question answering over linked data (qald-3): Lab overview. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 321–332. Springer, 2013.
6. Mark Alan Finlayson. Code for java libraries for accessing the princeton wordnet: Comparison and evaluation. 2013.
7. Daniel Gerber and Axel-Cyrille Ngonga Ngomo Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction ISWC*, 2011.
8. Ashwin Ittoo and Gosse Bouma. On learning subtypes of the part-whole relation: do not mix your seeds. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1328–1336, 2010.
9. Dekang Lin and Patrick Pantel. DIRT - discovery of inference rules of text. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM, 2001.
10. Vanessa Lopez, Miriam Fernández, Enrico Motta, and Nico Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.
11. Rahmad Mahendra, Lilian Wanzare, R Bernardi, A Lavelli, and B Magnini. Acquiring relational patterns from wikipedia: A case study. In *Proc. of the 5th Language and Technology Conference*, 2011.
12. John McCrae, Dennis Spohr, and Philipp Cimiano. Linking lexical resources and ontologies on the semantic web with lemon. In *The Semantic Web: Research and Applications*, pages 245–259. Springer, 2011.
13. John McCrae and Christina Unger. Design patterns for engineering the ontology-lexicon interface. In Paul Buitelaar and Philipp Cimiano, editors, *Towards the Multilingual Semantic Web*. Springer, 2014.
14. George Miller and Christiane Fellbaum. Wordnet: An electronic lexical database, 1998.
15. Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: A unified approach. *Proceedings of Transactions of the Association for Computational Linguistics (TACL)*, 2014.
16. Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

17. Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120, 2006.
18. Laurent Prévot, Chu-Ren Huang, Nicoletta Calzolari, Aldo Gangemi, Alessandro Lenci, and Alessandro Oltramari. Ontology and the lexicon: a multi-disciplinary perspective. In *Ontology and the Lexicon: A Natural Language Processing Perspective*, pages 3–24. Cambridge University Press, 2010.
19. Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. A new hybrid dependency parser for german. *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 115–124, 2009.
20. Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648. ACM, 2012.
21. Christina Unger, John McCrae, Sebastian Walter, Sara Winter, and Philipp Cimiano. A lemon lexicon for dbpedia. In *Proceedings of 1st International Workshop on NLP and DBpedia, October 21–25, Sydney, Australia*, 2013.
22. Marta Vila, Horacio Rodríguez, and M Antònia Martí. Wrpa: A system for relational paraphrase acquisition from wikipedia. *Procesamiento del lenguaje natural*, 45:11–19, 2010.
23. Sebastian Walter, Christina Unger, Philipp Cimiano, and Daniel Bär. Evaluation of a layered approach to question answering over linked data. In *The Semantic Web–ISWC 2012*, pages 362–374. Springer, 2012.