

Automatic Acquisition of Adjective Lexicalizations of Restriction Classes: A Machine Learning Approach

Sebastian Walter · Christina Unger · Philipp Cimiano

the date of receipt and acceptance should be inserted later

Abstract There is an increasing interest in providing common web users with access to structured knowledge bases such as DBpedia, for example by means of question answering systems. An essential task of such systems is transforming natural language questions into formal queries, e.g. expressed in SPARQL. To this end, such systems require knowledge about how the vocabulary elements used in the available ontologies and datasets are verbalized in natural language, covering different verbalization variants, possibly in multiple languages. An important part of such lexical knowledge is constituted by adjectives. In this paper we present and evaluate a machine learning approach to extract adjective lexicalizations from DBpedia. This is a challenge that so far has not been addressed. Our approach achieves an accuracy of 91.15% on a 10-fold cross validation regime. In addition to providing a first baseline system for the task of extracting adjective lexicalizations from DBpedia, we publish the extracted adjective lexicalizations in *lemon* format for free use by the community.

Keywords DBpedia · adjectives · ontology lexicon

1 Introduction

There is an increasing interest in supporting common web users in accessing structured knowledge bases such as DBpedia or Freebase, for example by means of question answering systems such as PowerAqua [16], TBSL [23], Bela [28], QAKiS [5] and FREyA [7]. An essential task of such systems is transforming natural language questions into formal queries, e.g. expressed in SPARQL. To this end, such systems require knowledge about how the vocabulary

elements used in the available ontologies and datasets are verbalized in natural language, covering different verbalization variants, possibly in multiple languages.

An important part of such lexical knowledge is constituted by adjectives. For example, the 250 training and test questions of the QALD-4 [24] benchmark¹ for question answering over DBpedia contain 76 different adjectives. Most of these adjectives are gradable (e.g. high) or intersective (e.g. Australian). While the former cannot be directly represented in OWL (see [18]), the latter denote simple restriction classes involving nominals. For example, Danish denotes the class $\exists \text{country}.\{\text{Denmark}\}$, female denotes the class $\exists \text{gender}.\{\text{Female}\}$, and Catholic denotes the class $\exists \text{religion}.\{\text{Catholic_Church}, \text{Catholicism}\}$.

Knowledge about such adjectives is crucial, for instance, when translating natural language questions such as 1 into SPARQL queries such as 2.

1. Which female Danish politicians are catholic?
2. PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT DISTINCT ?x WHERE {
 ?x rdf:type dbo:Politician .
 ?x dbo:country res:Denmark .
 ?x dbo:gender res:Female .
 { ?x dbo:religion res:Catholic_Church . }
UNION
 { ?x dbo:religion res:Catholicism . } }

In this example, the required mappings of adjectives in the natural language question in 1 to the SPARQL query presented in 2 are as presented in Table 1. (Note that the semantics of the adjective catholic is represented by a union of two basic graph patterns.) These mappings are

S. Walter
Semantic Computing Group, CITEC, Bielefeld University
E-mail: swalter@techfak.uni-bielefeld.de

¹ <http://www.sc.cit-ec.uni-bielefeld.de/qald/>

Adjective	Corresponding part in SPARQL query
female	?x dbo:gender res:Female .
Danish	?x dbo:country res:Denmark .
catholic	?x dbo:religion res:Catholic_Church .
catholic	?x dbo:religion res:Catholicism .

Table 1: Mapping from adjectives in the question in 1 to the corresponding parts of the SPARQL query in 2.

much harder to automatically find than, e.g., mapping the noun politician to the corresponding class `Politician`, in particular because the natural language string does not provide information on the required property.

As described in Section 5 in more detail, current approaches to learning lexicalizations, such as M-ATOLL [27], BOA [8] and WRPA [25], do not yet include methods for learning adjective lexicalizations. Therefore, the generated lexicons are necessarily incomplete and do not provide support when interpreting questions that include adjectives, such as the one above.

In this paper, we propose a machine learning approach to the extraction of adjective lexicalizations from a given knowledge base, in our case DBpedia, that is embedded in the framework M-ATOLL [27]. The paper is an extension of the work presented by Walter et al. [29], where only simple heuristics were used to generate adjective lexicalizations. This heuristic-based approach is subsumed by the approach we present in this paper, as the heuristics are included as features in the machine learning approach we present here.

Applying our approach to the whole of DBpedia 2014, we extract 15,380 adjective lexicalizations with an accuracy of 91.15%. We provide the extracted lexicalizations in *lemon* [19] format for free use:

https://github.com/ag-sc/matoll/public/june_2016/adjectives.ttl

The paper is structured as follows. In Section 2 we introduce our machine learning approach to extracting adjective lexicalizations, describing in particular the training dataset and the features used. In Section 3 we outline our experiments and results, providing insights into the impact of each feature on the task. We then describe the resulting adjective lexicalization dataset in Section 4. Before concluding, we discuss related work in Section 5.

2 Method

The main intuition underlying our approach to learn adjective lexicalizations with respect to a knowledge base is that an inspection of the objects occurring with a particular property often suggests relevant lexicalization candidates. The DBpedia property `gender`, for example, occurs very often with resources `Male` and `Female` as objects. Thus, male and female are obvious lexicalization

candidates for the restriction classes $\exists \text{gender}.\{\text{Male}\}$ and $\exists \text{gender}.\{\text{Female}\}$, respectively. Similarly, the property `country` occurs with objects like `Denmark` and `Germany`, which have related adjective forms `Danish` and `German`, which are lexicalizations of the restriction classes $\exists \text{country}.\{\text{Germany}\}$ and $\exists \text{country}.\{\text{Denmark}\}$, respectively.

In this section we explain in detail how our approach implements the extraction of adjectives from object occurrences and the decision which of the resulting candidates are valid lexicalizations.

2.1 Algorithm

Our approach comprises four steps, which are presented as pseudocode in Algorithm 1.

Algorithm 1 Approach to learning adjective lexicalizations.

```

1: for property  $p$  do
2:    $objects(p) \leftarrow \{o \mid (-, p, o) \in KB\}$ 
3:   for  $o$  in  $objects(p)$  do
4:     for terms  $a$  contained in  $label(o)$  do
5:       if  $a$  is an adjective then
6:          $v \leftarrow extractFeatures(a, p, o)$ 
7:         if  $classifyAsLexicalization(v)$  then
8:            $createAdjectiveEntry(a, p, o)$ 
9:         end if
10:      end if
11:    end for
12:  end for
13: end for

```

First, for a given property p , all RDF triples $(-, p, o)$ are retrieved from the knowledge base (see Algorithm 1 line 2). In particular, we are interested in the resource labels and literals appearing at the object position of property p . From now on, for some object o , we refer to these labels and literals as $label(o)$.

Then we extract adjective forms from the object labels by checking for each term occurring in the label whether WordNet [20] and DBnary [9] contain it as adjective, thereby building 3-tuples of the form (a, p, o) , where a represents the adjective form found in the label of the object o , and p is the property with which o occurred (see Algorithm 1 lines 4 and 5). For example, from the object label `Catholic Church` the term *catholic* is extracted. These extracted 3-tuples represent the candidate lexicalizations.

Subsequently, we classify all 3-tuples (a, p, o) with respect to whether they constitute a valid adjective lexicalization or not. To this end, we train a classifier using the features described in Section 2.3 below. In case the 3-tuple is classified positively, we finally create a lexical

entry in *lemon* format (see Algorithm 1 lines 7 and 8), using the template given in Figure 1.

In lines 1-7 of Figure 1, a resource of type `lemon:LexicalEntry` is created, where lines 3-7 specify the basic linguistic properties of this entry, such as its canonical form, its sense, its syntactic behavior and part of speech (`lexinfo:adjective` in our case). The canonical form indicates the written representation of the adjective (line 9), e.g. `Catholic`.

The property `lemon:sense` is used to express the meaning of the adjective with respect to the corresponding knowledge base by pointing to a restriction class (line 12), for example `∃religion.{Catholic_Church}`, which is defined in lines 15–17. In this case, `<p>` would be filled with the property `religion` and `<o>` would be filled with the object `Catholic_Church`.

Furthermore, each entry specifies two syntactic behaviors of the adjective:

- an `AdjectiveAttributiveFrame` (lines 19–20), capturing the attributive use of the adjective, as in *‘the catholic politician’*.
- an `AdjectivePredicativeFrame` (lines 22–23), capturing the predicative use of the adjective, e.g. in a copula construction such as *‘This politician is catholic’*.

The full entry for the adjective `Catholic` is shown in Figure 2.

We applied our approach to DBpedia 2014, processing every property and considering all objects of these properties. In the following sections we describe how the training data for the classifier has been created, and define the features used.

2.2 Training Dataset

Our training dataset consists of the five properties used by Walter et al. [29], i.e. `architecturalStyle`, `colour`, `geologicPeriod`, `militaryBranch` and `party`, together with eight additional properties that were randomly chosen to reduce the bias in our evaluation towards these manually selected properties. The resulting 12 DBpedia properties are shown in Table 2.

For each of these properties, we extracted all objects from DBpedia 2014 by means of the following SPARQL query:

```
SELECT ?o WHERE { _ <p> ?o . }
```

Given the resulting set $objects(p)$, we compute the frequency $freq(p, o)$ for each $o \in objects(p)$. For instance, for the property `religion`, there are 28,928 different objects; the top 10 most frequent ones are given in Table 3 together with the number of their occurrences. For the given properties, all unique 3-tuples (a, p, o) were extracted. The

Property (p)	Number of 3-tuples (a, p, o)
<code>religion</code>	427
<code>birthPlace</code>	402
<code>ethnicity</code>	326
<code>operatedBy</code>	322
<code>architecturalStyle</code>	257
<code>gameEngine</code>	132
<code>campusType</code>	117
<code>colour</code>	83
<code>geologicPeriod</code>	35
<code>gender</code>	15
<code>associatedRocket</code>	5
<code>foundedBy</code>	3
<code>elevation</code>	2

Table 2: Number of 3-tuples (a, p, o) for the annotated training properties.

Object	Frequency
<code>Catholic_Church</code>	6,422
<code>Islam</code>	1,959
<code>Christian</code>	1,042
<code>Baptists</code>	879
<code>Presbyterianism</code>	804
<code>Sunni_Islam</code>	781
<code>Hindu</code>	779
<code>Methodism</code>	772
<code>Episcopal_Church_(United_States)</code>	766

Table 3: Top 10 most frequent objects for the property `religion`.

number of 3-tuples extracted for each of the considered properties is shown in Table 2; it represents the amount of candidate adjective entries for that property.

The authors of this paper then annotated all 3-tuples with respect to whether they represent a valid adjective lexicalization or not, resulting in 1,074 valid lexicalizations and 1,051 non-valid ones.

In the next section we describe the features used to describe each 3-tuple.

2.3 Features

All features are defined over 3-tuples (a, p, o) , where a is an adjective that occurs in the label of an object o , and p is the property with which o occurs. To formally define the features, we introduce the following functions, where KB denotes the underlying knowledge base (in our case DBpedia).

- The function $label(o)$ returns the label of a given object o , e.g. for the resource `Catholic_Church` the label `Catholic Church` is returned. For literals, $label$ simply returns the literal value unchanged.
- The function $adjfreq_{KB}(p)$ computes the number of objects o for a given property p that ‘contain’ an

```

1 :LE_<a>
2   a lemon:LexicalEntry;
3   lemon:canonicalForm :LE_<a>_canonicalForm ;
4   lemon:sense :LE_<a>_sense ;
5   lemon:synBehaviour :LE_<a>_synBehaviour_1 ;
6   lemon:synBehaviour :LE_<a>_synBehaviour_2 ;
7   lexinfo:partOfSpeech lexinfo:adjective .
8
9 :LE_<a>_canonicalForm lemon:writtenRepresentation "<a>"@en ;
10
11 :LE_<a>_sense a lemon:Sense ;
12   lemon:reference LE_<a>_reference ;
13   lemon:isA _:b0 ;
14
15 :LE_<a>_reference a owl:Restriction ;
16   owl:onProperty <p> ;
17   owl:hasValue <o> .
18
19 :LE_<a>_synBehaviour_1 a lexinfo:AdjectiveAttributiveFrame;
20   lexinfo:attributiveArg _:b0.
21
22 :LE_<a>_synBehaviour_2 a lexinfo:AdjectivePredicativeFrame;
23   lexinfo:copulativeSubject _:b0.

```

Fig. 1: Template for adjective entry in *lemon* format, where <a> is a slot for the adjective form, <p> is a slot for the property, and <o> is a slot for the corresponding object.

```

1 :LE_Catholic
2   a lemon:LexicalEntry;
3   lemon:canonicalForm :LE_Catholic_canonicalForm;
4   lemon:sense :LE_Catholic_sense;
5   lemon:synBehaviour :LE_Catholic_synBehaviour_1;
6   lemon:synBehaviour :LE_Catholic_synBehaviour_2;
7   lexinfo:partOfSpeech lexinfo:adjective.
8
9 :LE_Catholic_canonicalForm lemon:writtenRepresentation "Catholic"@en ;
10
11 :LE_Catholic_sense a lemon:Sense;
12   lemon:reference LE_Catholic_reference;
13   lemon:isA _:b0.
14
15 :LE_Catholic_reference a owl:Restriction;
16   owl:onProperty <http://dbpedia.org/ontology/religion>;
17   owl:hasValue <http://dbpedia.org/resource/Catholic_Church>.
18
19 :LE_Catholic_synBehaviour_1 a lexinfo:AdjectiveAttributiveFrame;
20   lexinfo:attributiveArg _:b0.
21
22 :LE_Catholic_synBehaviour_2 a lexinfo:AdjectivePredicativeFrame;
23   lexinfo:copulativeSubject _:b0.

```

Fig. 2: Lexical entry for the property *religion* with the object *Catholic_Church*.

adjective, i.e. for which an adjective is an element in the tokenized sequence of $label(o)$:

$$adjfreq_{KB}(p) = \sum_{(s,p,o) \in KB} \begin{cases} 1, & \text{if } label(o) \text{ contains an adjective} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

– The function $adjfreq_{KB}(p, a)$ returns the frequency of the particular adjective a occurring in objects of property p :

$$adjfreq_{KB}(p, a) = \sum_{(s,p,o) \in KB} \begin{cases} 1, & \text{if } label(o) \text{ contains } a \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

For example, $adjfreq_{DBpedia}(\text{religion})$ is 12,720, i.e. 12,720 objects that occur as object of *religion* contain

an adjective, and $adjfreq_{DBpedia}(\text{religion}, \text{Hindu})$ is 779, i.e. the adjective Hindu occurs in 779 of those objects.

- Another function we will use is $freq_{KB}(p, o)$, which returns the overall frequency of a given object o in all RDF triples with a given property p :

$$freq_{KB}(p, o) = \sum_{(s,p,o') \in KB} \begin{cases} 1, & \text{if } o' = o \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

As an example consider the object `Greek_Orthodox_Church`, which occurs 174 times in the overall list of objects for the property `religion`, i.e. $freq_{DBpedia}(\text{religion}, \text{Greek_Orthodox_Church}) = 174$.

Analogously, we will use a function $pfreq_{KB}(p, p')$ that captures the frequency of a pattern p' in an object of all RDF triples with a given property p . (see 2.3.4 and 2.3.5).

In the following, we describe the features we used.

2.3.1 Normalized Object Frequency (NOF)

The *Normalized Object Frequency* counts the number of times that an object o occurs in objects of the property p , divided by the number of objects that actually contain an adjective:

$$NOF_{KB}(p, o) = \frac{freq_{KB}(p, o)}{adjfreq_{KB}(p)} \quad (4)$$

To illustrate this, consider the property `religion` with the object `Greek Orthodox Church`, which occurs 174 times in `religion`. The total number of objects containing an adjective is 12,720, therefore $NOF_{DBpedia}(\text{religion}, \text{Greek Orthodox Church}) = \frac{174}{12,720} = 0.013$.

2.3.2 Normalized Adjective Frequency (NAF)

The *Normalized Adjective Frequency* counts the number of times a particular adjective occurs in objects of a given property p , normalized by the total number of adjectives:

$$NAF_{KB}(a, p) = \frac{adjfreq_{KB}(p, a)}{adjfreq_{KB}(p)} \quad (5)$$

As an example, consider the adjective `Catholic`. It appears 5,545 times in objects of the property `religion`, and the objects containing an adjective is 25,587 adjectives in total. Therefore, $NAF_{DBpedia}(\text{Catholic}, \text{religion}) = \frac{5,545}{25,587} = 0.21$. This means that out of all adjectives occurring in objects of the property `religion`, `Catholic` occurs in 21% of the cases.

2.3.3 Adjective Ratio (AR)

The *Adjective Ratio* computes the number of objects of a property p that contain an adjective, normalized with respect to the total number of objects:

$$AR_{KB}(p) = \frac{adjfreq_{KB}(p)}{|objects_{KB}(p)|} \quad (6)$$

For the property `religion`, for example, 12,720 adjectives are found in 25,587 objects. Therefore, $AR_{DBpedia}(\text{religion}) = \frac{12,720}{25,587} = 0.49$, which means that only roughly half of its objects contain an adjective.

2.3.4 Pattern Ratio (PR)

We transform the label of each object $o \in objects(p)$ into a pattern by replacing the corresponding adjective a by the string *ADJ*. For the property `religion`, for example, this yields the patterns shown in Table 4. Therefore, the function $pattern(o)$, which is used below, returns for a given object o the corresponding pattern, e.g. for the object `Greek Orthodox Church` the pattern `Greek ADJ Church` is returned.

For each triple (a, p, o) , we record the frequency of the pattern extracted from o normalized by the sum of the frequencies of all patterns appearing in objects of that property:

$$PR_{KB}(p, o) = \frac{pfreq_{KB}(p, pattern(o))}{\sum_{o' \in objects(p)} pfreq_{KB}(p, pattern(o'))} \quad (7)$$

The *Pattern Ratio* thus describes the number of occurrences of the pattern of a particular object compared to the sum of the occurrences of the patterns of all objects.

2.3.5 Part-of-Speech Pattern Ratio (POSPR)

Next we define an extension of the pattern ratio above, replacing all terms in an object label by their part-of-speech tag. The label `Catholic Church`, for example, would resolve into the pattern `JJ NN`. We refer to this pattern of an object o as *POSPattern*(o) and define the *Part-of-Speech Pattern Ratio* as follows:

$$POSPR_{KB}(p, o) = \frac{pfreq_{KB}(p, POSPattern(o))}{\sum_{o' \in objects(p)} pfreq_{KB}(p, POSPattern(o'))} \quad (8)$$

It thus describes the number of occurrences of the part-of-speech pattern of some object compared to the sum of occurrences of the part-of-speech patterns of all objects.

Pattern	Frequency
<i>ADJ</i> Church	5,721
<i>ADJ</i>	2,570
<i>ADJ</i> Islam	772
Eastern <i>ADJ</i> Church	619
The Church of Jesus Christ of <i>ADJ</i> Saints	300
United <i>ADJ</i> Church	231
Greek <i>ADJ</i> Church	137
Serbian <i>ADJ</i> Church	129
<i>ADJ</i> Church of Canada	120
Southern <i>ADJ</i> Convention	106
Armenian <i>ADJ</i> Church	76
Dutch <i>ADJ</i> Church	72
<i>ADJ</i> Buddhism	70
Roman <i>ADJ</i> Kirk	64
<i>ADJ</i> Christianity	59
<i>ADJ</i> Religion	58
<i>ADJ</i> Judaism	57
<i>ADJ</i> Universalism	53
Seventh-day <i>ADJ</i> Church	51
Russian <i>ADJ</i> Church	42

Table 4: Top 20 most frequent patterns for the property `religion` together with their frequency.

2.3.6 Position Features (*AP*, *AFP*, *ALP*)

This group of features encodes information about the position of the adjective a in the label of an object o .

For each 3-tuple (a, p, o) , we encode the position at which a occurs in the label of o as a feature *AP*. For example, the adjective *Catholic* occurs in the label *Catholic Church* at position 1, while the adjective *Orthodox* occurs in *Romanian Orthodox Church* at position 2.

As specific cases, the Boolean feature *Adjective First Position* (*AFP*) indicates whether the adjective occurs as the first word in the label of the object, and the Boolean feature *Adjective Last Position* (*ALP*) indicates whether the adjective occurs as the last word in the label of the object.

2.3.7 Normalized Levenshtein Distance (*NLD*)

This feature computes the Normalized Levenshtein [15] distance between an adjective a and the label of an object o :

$$NLD(a, o) = \frac{\text{LevenshteinDistance}(a, \text{label}(o))}{\max(\text{length}(a), \text{length}(\text{label}(o)))} \quad (9)$$

2.3.8 Character n -grams

For each adjective a in (a, p, o) , we extract its character trigram and bigram suffixes. For example, for the adjective *Hindu* we extract the character trigram *-ndu* and the character bigram *-du*. For each such character n -gram, we construct one Boolean feature recording whether the n -gram occurs in the particular adjective a or not, yielding a total of

368 trigram and 160 bigram features for each adjective in our dataset.

2.3.9 Part-of-Speech Pattern (*PP*)

For each adjective a in (a, p, o) , we extract the part-of-speech pattern $POSPattern(o)$. Overall, the data contains 428 different patterns. We include a feature for each of these 398 patterns indicating if the object o matches the given pattern.

2.3.10 Part-of-Speech Adjective Pattern (*PAP*)

For each adjective a in (a, p, o) we replace a in the label of o by the string *ADJ* and replace all other terms by their part-of-speech tag. For the adjective *Catholic*, for example, the object label *Catholic Church* is transformed into the pattern *ADJ NN*. We consider 398 patterns and as for the *PP* feature we include a feature for each of these 428 patterns indicating if the object o matches the given pattern.

3 Evaluation

In our experiments we test eight different classifiers: Decision Trees, Random Trees, Random Forests, Support Vector Machines with Sequential Minimal Optimization, Logistic Regression, Bayes Nets, Voted Perceptrons and Decision Tables, as implemented in WEKA 3.8 [10]. For each we perform an exhaustive search in the space of all feature groups to determine the best set of features for the task. As Random Forests achieved the best results, we report the results of the best feature set for this classifier. Then we compare the results for this feature set for all classifiers, see Section 3.2.

We use a ten-fold cross validation setting to perform the feature selection. We do not evaluate the selected features on additional, unseen data, but we regard our results as clear indicators for the features that work well on the task, given that the results are similar across classifiers, with 91.15% the top accuracy (Random Forest) and 82.68% (Decision Table) the lowest.

In addition, we perform a ten-fold cross validation to compute the accuracy of each feature in isolation, see Section 3.1.

The training data is as described in Section 2.2 and consists of the list of annotated 3-tuples (a, p, o) . With 1,074 positive entries and 1,051 negative entries, the dataset is balanced. We use the cross validation strategy built in into WEKA, thus each triple (a, p, o) is contained in exactly one fold and triples sharing one element are not necessarily in the same fold.

Feature	Accuracy	Variance	Standard Deviation
NAF	83.48	5.85	2.42
Trigrams	80.37	6.2	2.49
Bigrams	76.14	8.46	2.91
PR	75.72	9.24	3.04
POSPR	75.72	9.24	3.04
AR	75.67	9.24	3.04
PAP	73.08	10.11	3.18
NOF	71.30	9.79	3.13
PP	71.29	8.52	2.92
NLD	70.73	11.35	3.37
AP	59.52	7.34	2.71
AFP	57.93	10.89	3.30
ALP	54.21	9.98	3.16

Table 5: Accuracy (in %) per feature.

3.1 Accuracy of Features in Isolation

We evaluated all 13 feature groups individually in a ten-fold cross validation setting. The results are presented in Table 5, starting with the feature group with the highest accuracy. Additionally, for each feature, the variance and standard deviation is given.

The feature with the highest impact is the *Normalized Adjective Frequency (NAF)*, reaching an accuracy of 83.48% with a standard deviation of 2.42, followed by the group of trigram features encoding the last three characters of an adjective, reaching an accuracy of 80.37% with a standard deviation of 2.49. Bigrams lead to an accuracy of 76.14% with a standard deviation of 2.91. The features with the lowest impact are the features which encode the position of the adjective, namely *AP*, *AFP* and *ALP*, with an accuracy of 59.52%, 57.93% and 54.21%, respectively. From these three, the *AP* feature is the best, as it encodes the direct position of the adjective and therefore indirectly subsumes the other two position features.

The adjective ratio *AR* was the main feature in our previous work [29], representing the *adjectiveness* of the objects encountered for a property. It reaches an accuracy of 75.67% and ranks highest when using a Decision Tree classifier.

3.2 Feature Analysis

In order to analyze the impact of the features also in combination with other features, we perform an exhaustive search over the possible feature combinations. This amounts to an evaluation of 8,192 feature combinations.² Each combination was separately evaluated using our ten-fold cross validation task, in order to determine the most effective feature combination. Table 6 reports the ten best

² Which can be downloaded at http://sebastianwalter.org/downloads/dblexipedia/jods_arff_files.tar.gz as *.arff* files.

Feature Combinations	Acc.
PR, ALP, Trigrams, AFP, AP, AR, NAF	91.15
PP, NOF, ALP, Trigrams, AR, NAF, POSPR, Bigrams	90.87
PP, PR, ALP, Trigrams, AP, AR, NAF, Bigrams	90.82
PP, PR, Trigrams, NAF, POSPR, Bigrams	90.78
PP, PR, ALP, Trigrams, AP, AR, NAF, POSPR	90.73
PP, NOF, ALP, Trigrams, AFP, AR, NAF, POSPR, Bigrams	90.72
PP, PR, Trigrams, AR, NAF, POSPR, Bigrams	90.68
PP, PR, NOF, ALP, Trigrams, AR, NAF, Bigrams	90.63
PP, Trigrams, AP, AR, NAF, POSPR, Bigrams	90.63
PP, PR, ALP, Trigrams, NAF	90.59

Table 6: Accuracy (in %) for the top 10 feature combinations.

Classifier	Accuracy (best)	Accuracy (all)
Random Forest	91.15	89.83
Decision Tree (J48)	87.29	86.63
Random Tree	86.68	81.97
SVM with SMO	85.64	86.50
Logistic Regression	85.27	83.95
Bayes Net	84.80	84.80
Voted Perceptron	83.62	84.09
Decision Table	82.68	81.60

Table 7: Accuracy (in %) for all classifiers, using the best feature combination (PR, ALP, Trigrams, AFP, AP, AR) as well as all features.

combinations, all of which outperform each feature in isolation. The highest accuracy is 91.15%, achieved by a Random Forest classifier and using only 7 out of the 13 feature groups; the lowest accuracy is 54%.

In Table 7 we report the results achieved by all eight classifiers, once with the best feature combination (PR, ALP, Trigrams, AFP, AP, AR, see Table 6) and once using all features, showing that the Random Forest classifier achieves the best results in both cases.

It is interesting to highlight that the optimal feature combination for the Random Forest also represents the optimal feature combination for four out of the other seven classifiers (Decision Tree, Random Tree, Logistic Regression, Decision Table). The Bayes Net classifier showed no difference in accuracy, while the SVM with SMO and the Voted Perceptron yield a small increase of accuracy when using all features.

3.3 Discussion

In this section we discuss the results for six properties in more detail – three properties from the training set (*colour*, *gender*, *religion*) and three other properties (*governmentType*, *nationality*, *originalLanguage*) in order to verify the generalization of our approach to properties not seen during training.

Table 8 shows the accuracy and the number of created lexical entries for the above mentioned six properties. For each property, all lexical entries generated by our

Property	Accuracy	Created entries
colour	84	71
gender	75	12
religion	80	271
governmentType	67	88
nationality	88	361
originalLanguage	100	15

Table 8: Manually evaluated accuracy (in %) on six example properties.

approach were manually evaluated, deciding whether they represent valid lexicalizations. The results show that our approach extracts reasonable lexicalizations, with an accuracy ranging from 67 % for `governmentType` to 100 % for `originalLanguage`.

In Table 9 we show examples of the extracted adjectives for each property from Table 8, together with the object from which the adjective was extracted.

An adjective that is frequently proposed by our method is the adjective `united`, occurring in object labels such as `United States` (with the property `governmentType`) or `United Church of Christ` (with the property `religion`). Such cases are difficult to automatically distinguish from cases where the lexicalizations are correct. In some cases it is, in fact, even hard to decide as a human whether the adjective is indeed a suitable lexicalization. For example, the property `nationality` often occurs with objects describing languages, such as `French language` and `Kurdish languages`.

We assume that we could improve the results by taking into account the range of a given property, thus learning that particular adjective lexicalizations make sense only for particular types of objects.

Another example that demonstrates the limitations of our approach is the property `birthPlace`. For this property a total of 2,519 lexical entries were extracted; some example lexicalizations together with the corresponding object label are presented in Table 10. The problem is that there is a high number of adjectives in the object labels but most of them should be filtered out – a case that the classifiers did not encounter clearly enough during training.

Due to the nature of our approach, no adjective lexicalizations are extracted for datatype properties with literals other than strings. For properties such as `birthDate`, for example, adjective lexicalizations such as `young` or `old` would be relevant but fall outside the scope of our approach.

4 Resulting Dataset

We applied the trained Random Forest model to all 2,796 properties in DBpedia 2014. This resulted in a *lemon* lexicon with 15,380 adjective entries, with an average of 14.2 entries per property.

We make three datasets available: the file which was used to train the Random Forest model (`adjective.arff`), the model itself (`adjective.model`), and the resulting *lemon* lexicon in Turtle syntax (`adjectivexicon.ttl`). All three files can be found at the following location:

https://github.com/ag-sc/matoll/tree/master/public/june_2016/

The implementation of our approach is open source and can be accessed in the `LabelApproach` folder of the M-ATOLL repository on GitHub:

<https://github.com/ag-sc/matoll/>

5 Related Work

There has been a substantial body of work on lexical acquisition from corpora [2, 22]. One important task in lexical acquisition is the extraction of verb frames and subcategorization frames from corpora (see, e.g., [21]). There has also been a lot of work on clustering adjectives by their semantics [3, 6]. Further work has addressed the extraction of hyponyms and hypernyms from corpora [14], and of identifying the particular meaning of adjectives in adjective-noun phrase compounds, e.g. relying on topic models [12, 11, 13]. Adjectives have also been exploited as attributes to cluster nouns [1].

However, there has been little work so far on identifying the meaning of adjectives with respect to existing large ontologies such as DBpedia. For example, Maillard et al. [17] have proposed a tensor-based framework using skip-gram models to represent the meaning of adjectives. This approach induces a latent semantics from a corpus rather than trying to capture the semantics of an adjective with respect to existing ontologies.

In contrast, Boleda et al. [4] create an ontology lexicon for Catalan, starting from a pre-defined list of adjectives, which are then classified into unary and binary (relational) adjectives.

In this paper, we have extended our ontology lexicalization approach M-ATOLL [26, 27] to adjectives. This makes it more extensive than other existing ontology lexicalization approaches, such as BOA [8] and WRPA [25], which do not consider adjective lexicalizations.

Both M-ATOLL and BOA start from RDF triples from a linked data source and a corresponding text corpus, and extract property lexicalizations from sentences containing the labels of the subjects and objects occurring with the property in question. BOA and WRPA extract the strings between the occurrences of these labels. In contrast, M-ATOLL considers the dependency parse structure of the relevant sentences. Both approaches have been applied to DBpedia, using an English Wikipedia text corpus.

Property	Adjective	Object
colour	red	Red
colour	purple	Purple
colour	cyan	Cyan
colour	ivory	Ivory_(color)
gender	female	Female
gender	male	Male
gender	unisex	Unisex
gender	mixed-sex	Mixed-sex_education
religion	christian	Saint.Thomas.Christian.churches
religion	catholic	Roman.Catholic.Diocese.of.Saskatoon
religion	apostolic	New.Apostolic.Church
religion	orthodox	Georgian.Orthodox.Church
governmentType	constitutional	Constitutional.monarchy
governmentType	multi-party	Multi-party_system
governmentType	federation	Federation
governmentType	district	District_municipality
nationality	scottish	Scottish_people
nationality	portuguese	Portuguese_people
nationality	peruvian	Peruvian_people
nationality	sikh	Sikh
originalLanguage	latin	Latin
originalLanguage	hindi	Hindi
originalLanguage	greek	Ancient_Greek
originalLanguage	english	English_language

Table 9: Example adjectives extracted for the six example properties.

Lexicalization	Object Label
spanish	Spanish Empire
french	French Chad
worthy	Martyr Worthy
national	National City, California
little	Little Plumstead
green	Birches Green

Table 10: Example lexicalizations for the property `birthPlace` together with the object label from which the adjective was extracted.

M-ATOLL additionally uses the Spanish and German Wikipedia to extract lexicalizations in those languages. Similarly, WRPA extracts lexicalization patterns from the English and Spanish Wikipedia.

6 Conclusion and Future Work

In this paper we presented an approach for extracting adjective lexicalizations with respect to an ontology by analyzing the labels of objects occurring in a given dataset. We showed that the features we selected for this task achieve a reasonable accuracy. We thus introduced a relatively low-cost and accurate method for inducing an adjective lexicon, which can be straightforwardly applied to other ontologies and languages.

One aspect of future work is thus to evaluate how well our approach works for languages other than English. Since the features we use are language independent (possibly with exception of the part-of-speech tags), it would be very

interesting to see whether the model trained on English would also work reasonably well for other languages, in particular German and Spanish, which are covered by M-ATOLL already.

Another important aspect is to indirectly evaluate the resulting lexicon on different tasks such as question answering or natural language generation, as an extensive manual evaluation is not feasible for this large amount of generated data.

Acknowledgments

This work was supported by the Cluster of Excellence Cognitive Interaction Technology *CITEC* (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

References

1. Almuhareb, A., Poesio, M.: Attribute-Based and Value-Based Clustering: An Evaluation. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
2. Boguraev, B., Pustejovsky, J. (eds.): Corpus Processing for Lexical Acquisition. MIT Press, Cambridge, MA, USA (1996)

3. Boleda Torrent, G., Alonso i Alemany, L.: Clustering Adjectives for Class Acquisition. In: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics (EACL '03), Budapest, Hungary (2003)
4. Boleda Torrent, G., Badia, T., Batlle, E.: Acquisition of Semantic Classes for Adjectives from Distributional Evidence. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), Geneva, Switzerland (2004)
5. Cabrio, E., Cojan, J., Aprosio, A.P., Magnini, B., Lavelli, A., Gandon, F.: QAKiS: an open domain QA system based on relational patterns. In: Proceedings of the 11th International Semantic Web Conference (ISWC 2012), Boston, USA (2012)
6. Chen, K.h., Chen, H.H.: Corpus-Based Analyses of Adjectives: Automatic Clustering. In: Proceedings of the International Conference on Quantitative Linguistics, Moscow, Russia (1994)
7. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. In: Proceedings of the 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece (2010)
8. Gerber, D., Ngonga Ngomo, A.C.: Bootstrapping the linked data web. Proceedings of the 1st Workshop on Web Scale Knowledge Extraction, workshop co-located with the 10th International Semantic Web Conference (ISWC 2011), Bonn, Germany (2011)
9. Gilles, S.: Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web* **6**(4), 355–361 (2015)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)
11. Hartung, M., Frank, A.: A Semi-supervised Type-based Classification of Adjectives: Distinguishing Properties and Relations. In: Proceedings of the 7th international conference on Language Resources and Evaluation (LREC), Malta (2010)
12. Hartung, M., Frank, A.: A Structured Vector Space Model for Hidden Attribute Meaning in Adjective-noun Phrases. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10), Beijing, China (2010)
13. Hartung, M., Frank, A.: Exploring Supervised LDA Models for Assigning Attributes to Adjective-noun Phrases. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11), Edinburgh, UK (2011)
14. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the 14th Conference on Computational Linguistics (COLING'92), Nantes, France (1992)
15. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: *Soviet physics doklady*, vol. 10, pp. 707–710 (1966)
16. Lopez, V., Fernández, M., Motta, E., Stieler, N.: PowerAqua: supporting users in querying and exploring the Semantic Web content. *Semantic Web* **3**, 249–265 (2012)
17. Maillard, J., Clark, S.: Learning Adjective Meanings with a Tensor-Based Skip-Gram Model. *CoNLL 2015* p. 327 (2015)
18. McCrae, J., Quattri, F., Unger, C., Cimiano, P.: Modelling the Semantics of Adjectives in the Ontology-Lexicon Interface. In: Proceedings of the Cognitive Aspects of the Lexicon (CogAlex), workshop co-located with the 25th International Conference on Computational Linguistics (COLING 2014), Dublin, Ireland (2014)
19. McCrae, J., Spohr, D., Cimiano, P.: Linking lexical resources and ontologies on the semantic web with lemon. In: *The Semantic Web: Research and Applications*, pp. 245–259. Springer, Berlin, Heidelberg (2011)
20. Miller, G.A.: WordNet: A Lexical Database for English. *Communications of the ACM* **38**(11), 39–41 (1995)
21. Preiss, J., Briscoe, T., Korhonen, A.: A System for Large-Scale Acquisition of Verbal, Nominal and Adjectival Subcategorization Frames from Corpora. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic (2007)
22. Pustejovsky, J.: The Acquisition of Lexical Semantic Knowledge from Large Corpora. In: Proceedings of the Workshop on Speech and Natural Language, workshop co-located with the Association for Computational Linguistics (ACL), Harriman, New York, USA (1992)
23. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.C., Gerber, D., Cimiano, P.: Template-based Question Answering over RDF Data. In: Proceedings of the 21st International Conference on World Wide Web (WWW), Lyon, France (2012)
24. Unger, C., Forascu, C., Lopez, V., Ngonga Ngomo, A.C., Cabrio, E., Cimiano, P., Walter, S.: Question Answering over Linked Data (QALD-4). In: Working Notes for the CLEF 2014 Conference, Sheffield, UK (2014)
25. Vila, M., Rodríguez, H., Martí, M.A.: WRPA: A system for relational paraphrase acquisition from Wikipedia. *Procesamiento del lenguaje natural* **45**, 11–19 (2010)

26. Walter, S., Unger, C., Cimiano, P.: ATOLL - A framework for the automatic induction of ontology lexica. *Data & Knowledge Engineering* **94, Part B**, 148 – 162 (2014). Special issue following the 18th International Conference on Applications of Natural Language Processing to Information Systems (NLDB'13)
27. Walter, S., Unger, C., Cimiano, P.: M-ATOLL: A Framework for the Lexicalization of Ontologies in Multiple Languages. In: *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*, Riva del Garda, Italy (2014)
28. Walter, S., Unger, C., Cimiano, P., Bär, D.: Evaluation of a Layered Approach to Question Answering over Linked Data. In: *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, Boston, USA (2012)
29. Walter, S., Unger, C., Cimiano, P., Lanser, B.: Automatic acquisition of adjective lexicalizations of restriction classes. In: *Proceedings of the 2nd International Workshop on NLP and DBpedia, co-located with the 13th International Semantic Web Conference (ISWC 2014)*, Riva del Garda, Italy (2014)