Generating Lexicalization Patterns for Linked Open Data

Rivindu Perera and **Parma Nand**
Auckland University of Technology
Auckland, New Zealand
{rperera, pnand}@aut.ac.nz

Abstract

The concept of Linked Data has attracted increased interest in recent times due to its free and open availability and the sheer of volume. We present a framework to generate patterns which can be used to lexicalize Linked Data. We use DBpedia as the Linked Data resource which is one of the most comprehensive and fastest growing Linked Data resource available for free. The framework incorporates a text preparation module which collects and prepares the text after which Open Information Extraction is employed to extract relations which are then aligned with triples to identify patterns. The framework also uses lexical semantic resources to mine patterns utilizing VerbNet and WordNet. The framework achieved 70.36% accuracy and a Mean reciprocal Rank value of 0.72 for five DBpedia ontology classes generating 101 lexicalizations.

1 Introduction

Semantic web continues to grow rapidly in various forms. Two key areas that recent semantic web researches have focused on are enrichment of Linked Data resources and using these resources in different applications.

DBpedia, Freebase, and YAGO[1] are frontiers in Linked Data area. The Linked Data is represented as triples (a data structure in the form of ⟨subject, predicate, object⟩) using Resource Description Framework (RDF). As Linked Data concept moves forward, there is also a need to utilize this data in applications. A major area that requires Linked Data is Natural Language Processing (NLP) and applications such as Question Answering (QA) (Perera, 2012a; Perera, 2012b). A drawback of Linked Data is that it lacks the linguistic information which can be used to turn them back to a natural textual format.

Generating linguistic structures and choosing words to communicate a particular abstract representation (e.g., triple) is referred to as lexicalization which is a subtask in Natural Language Generation. The work described in this paper is a part of our NLG project[2] currently under way (Perera and Nand, 2014a; Perera and Nand, 2014b; Perera and Nand, 2014c). The framework presented in this paper uses DBpedia as the Linked Data resource and lexicalization is presented as the mining best available pattern to generate a natural language representation for the triple being considered.

The remainder of the paper is structured as follows. Section 2 presents related work in the area of lexicalization. In Section 3 we describe the proposed framework in detail. Section 4 presents the experiments used to validate the framework. Section 5 concludes the paper with an outlook on future work.

2 Related work

Duma and Klein (2013) present an approach to extract templates to verbalize triples using a heuristic. The main drawbacks noticed in this model are the ignorance of additional textual resources and less consideration on the cohesive pattern generation

Lemon model (Walter et al., 2013) extracts lexicalizations for DBpedia using dependency patterns extracted from Wikipedia sentences. However, the initial experiments we performed have shown that this approach fails completely when provided with sentences with grammatical conjunctions.

Ell and Harth (2014) introduce the language in-

[1] dbpedia.org, freebase.com, mpi-inf.mpg.de/yago/

[2] http://rivinduperera.com/information/realtextlex

Proceedings of the Second Workshop on Natural Language Processing and Linked Open Data, pages 2–5,
Hissar, Bulgaria, 11 September 2015.

dependent approach to generate RDF verbalization templates. This model utilizes the maximal sub-graph pattern extraction model. However, in our approach the Open Information Extraction (OpenIE) is utilized to get more coherent lexicalization patterns (Perera and Nand, 2015a; Perera and Nand, 2015b).

3 RealText$_{lex}$ framework

Fig. 1 depicts the high-level overview of the process of generating lexicalization patterns in the proposed framework. The process starts with a given DBpedia ontology class (e.g., person, organization, etc.). The following sections explains the process in detail.

3.1 Candidate sentence extraction

The objective of candidate sentence extractor is to identify potential sentences that can lexicalize a given triple. The input is taken as a collection of co-reference resolved sentences and a set of triples. This unit firstly verbalizes the triples using a set of rules. Then each sentence is analysed to check either complete subject (s), the object (o) or the predicate (p) are mentioned in the sentence (S). This sentence analysis assigns a score to each sentence based on presence of a triple. The score is the ratio of subject, predicate and object present in the sentence.

3.2 Open Information Extraction

Once the candidate sentences are selected for each triple, we then extract relations from these candidate sentences employing Open IE. The Open IE (Etzioni et al., 2008) essentially focuses on domain independent relation extraction and predominantly targets the web as a corpus for deriving the relations. The framework proposed in this paper uses textual content extracted from the web which works with a diverse set of domains. Specifically, the framework uses Ollie Open IE system[3] for relation extraction. This module associates each relation with the triple and outputs a triple-relations collection. A relation is composed of first argument (arg1), relation (rel), and second argument (arg2).

3.3 Pattern processing and combination

This module generates patterns from aligned relations in Section 3.2. In addition to these patterns, verb frame based patterns are also determined and added to the pattern list.

3.3.1 Relation based patterns

Based on the aligned relations and triples, a string based pattern is generated. These string based patterns can get two forms as shown in Fig. 2 for two sample scenarios. The subject and object are denoted by symbols *s?* and *o?* respectively.

3.3.2 Verb frame based patterns

The framework utilizes two lexical semantic resources, VerbNet and WordNet to mine patterns. Currently, the framework generates only one type of pattern (`s? Verb o?`), if the predicate is a verb and if that verb has the frame {*Noun phrase, Verb, Noun phrase*} in either VerbNet or WordNet.

3.3.3 Property based patterns

The predicates which cannot be associated with a pattern in the above processes described in Section 3.3.1 and Section 3.3.2 are properties belonging to the DBpedia resources selected. The left over predicates are assigned a generic pattern (`s? has ⟨predicate⟩ of o?`) based on the specific predicate.

3.4 Pattern enrichment

Pattern enrichment adds two types of additional information; grammatical gender related to the pattern and multiplicity level associated with the determined pattern. When searching a pattern in the lexicalization pattern database, these additional information is also mined in the lexicalization patterns for a given predicate of an ontology class.

3.4.1 Grammatical gender determination

The lexicalization patterns can be accurately reused later only if the grammatical gender is recorded with the pattern. For example, consider triple, ⟨`Walt Disney, spouse, Lillian Disney`⟩ and lexicalization pattern, "`s? is the husband of o?`". This pattern cannot be reused to lexicalize the triple ⟨`Lillian Disney, spouse, Walt Disney`⟩, because the grammatical gender of the subject is now different, even though the property (spouse) is same in both scenarios. The framework uses three types of grammatical gender types (male, female, neutral) based on the triple subject and it is determined by DBpedia grammatical gender dataset (Mendes et al., 2012).

[3] knowitall.github.io/ollie/

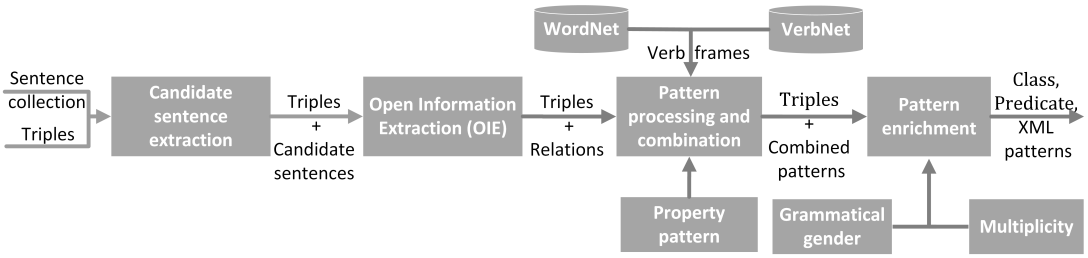

Figure 1: Schematic representation of the complete framework

- ⟨Walt Disney, birth date, 1901-12-05⟩
  - `arg1:` Walt Disney, `rel:` was born on, `arg2:` December 5, 1901 `pattern:` s? was born on o?
- ⟨Walt Disney, designer, Mickey Mouse⟩
  - `arg1:` Mickey Mouse, `rel:` is designed by, `arg2:` Walt Disney `pattern:` o? is designed by s?

Figure 2: Basic patterns generated for two sample triples. *s?* and *o?* represent subject and object respectively.

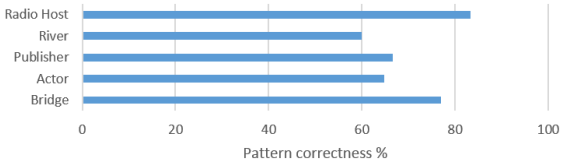

Figure 3: Analysis of syntactic correctness of the extracted patterns

3.4.2 Multiplicity determination

In DBpedia page for Nile River has three countries listed under the predicate "country" because it does not belong to one country, but flows through these countries. However, East River belongs only to United States. The lexicalization patterns generated for these two scenarios will also be different and cannot be shared. For example, lexicalization pattern for Nile river will in the form of "`s? flows through o?`" and for East River it will be like "`s? is in o?`". To address this variation, our framework checks whether there are multiple object values for the same subject and predicate, then it adds the appropriate property value (multiple/single) to the pattern.

4 Experimental framework

4.1 Experimental settings and results

Table 1 shows the summary of the breakdown of the results for pattern extraction. The last 5 columns of the table also shows the results for the pattern enrichment modules. To get a clear idea on the accuracy of the framework, we checked how many syntactically correct lexicalization patterns appear as the highest ranked pattern for the given predicate. In this context syntactic correctness was considered as being both grammatically accurate and coherent. The results of this evaluation is shown in Fig. 3 for each of the ontology classes.

Since, the framework ranks lexicalization patterns using a scoring system, we considered it as a method that provides a set of possible outputs. We decided to get a statistical measurement incorporating Mean Reciprocal Rank (MRR) as shown below to compute the rank of the first correct pattern of each predicate in each ontology class.

$$MRR = \frac{1}{|P|}\sum_{i=1}^{|P|}\frac{1}{rank_i} \quad (1)$$

where P and $rank_i$ represent predicates and the rank of the correct lexicalization for the i^{th} predicate respectively. Table 2 depicts the MRR results for the 5 ontology classes being considered.

Table 3 shows a statistical summary of proposed approach.

4.2 Observations and discussions

The following observations can be made based on the results of the experiment. Fig. 3 shows that our framework has achieved 70.36% average accuracy for 5 ontology classes where the lowest accuracy was reported as 60%. This evaluation does not take into account the rank of the correct lexicalization patterns and measures the number of correct patterns present in the extracted set of patterns. On the other hand, MRR based evaluation

Table 1: Results of the pattern extraction module

Ontology class	Relational patterns	Frame patterns	Property patterns	Pattern enrichment				
				Multiplicity		Grammatical gender		
				Multiple	Single	Male	Female	Neutral
Bridge	272	8	9	163	126	0	0	289
Actor	422	0	16	369	69	400	22	16
Publisher	39	1	4	32	12	0	0	44
River	157	2	10	158	11	0	0	169
Radio Host	30	1	1	14	18	0	0	32

Table 2: Mean Reciprocal Rank analysis for ranked lexicalization patterns

	Bridge	Actor	Publish	River	Radio Host
MRR	0.77	0.69	0.72	0.61	0.83

Table 3: Statistics of evaluation of proposed approach

Candidate templates	Lexicalizations	Accuracy
393	101	70.36%

provides an detailed look at ranking of the first correct lexicalization. Average MRR value of 0.724 achieved for 5 ontology classes. Finally, based on the comparison in Table 3, it is clear that proposed approach in this paper has advanced the way of deriving lexicalizations by generating reasonable number of valid patterns and with a higher accuracy.

5 Conclusion and future work

This paper presented a framework to generate lexicalization patterns for DBpedia triples using a pipeline of processes. The pipeline starts with ontology classes which is then used to mine patterns aligning triples with relations extracted from sentence collections from the web. The framework generated patterns were human-evaluated and showed an accuracy of 70.36% and a MRR of 0.72 on test dataset. In future, we aim to target on expanding the test collection to build a reasonable sized lexicalization pattern database for DBpedia.

References

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *IWCS-2013*.

Basil Ell and Andreas Harth. 2014. A language-independent method for the extraction of rdf verbalization templates. In *INLG-2014*.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51.

Pablo N. Mendes, Max Jakob, and Christian Bizer. 2012. DBpedia for NLP: A Multilingual Cross-domain Knowledge Base. In *LREC-2012*.

Rivindu Perera and Parma Nand. 2014a. Interaction history based answer formulation for question answering. In *KESW-2014*, pages 128–139.

Rivindu Perera and Parma Nand. 2014b. Real text-cs - corpus based domain independent content selection model. In *ICTAI-2014*, pages 599–606.

Rivindu Perera and Parma Nand. 2014c. The role of linked data in content selection. In *PRICAI-2014*, pages 573–586.

Rivindu Perera and Parma Nand. 2015a. A multi-strategy approach for lexicalizing linked open data. In *CICLing-2015*, pages 348–363.

Rivindu Perera and Parma Nand. 2015b. Realtext-lex: A lexicalization framework for linked open data. In *ISWC-2015 Demonstration*.

Rivindu Perera. 2012a. Ipedagogy: Question answering system based on web information clustering. In *T4E-2012*.

Rivindu Perera. 2012b. *Scholar: Cognitive Computing Approach for Question Answering*. Honours thesis, University of Westminster.

Sebastian Walter, Christina Unger, and Philipp Cimiano. 2013. A corpus-based approach for the induction of ontology lexica. In *NLDB-2013*, Salford.