Lexicalizing DBpedia with Realization Enabled Ensemble Architecture: RealText$_{lex2}$ Approach

Rivindu Perera, Parma Nand, and Gisela Klette

School of Computer and Mathematical Sciences,
Auckland University of Technology, New Zealand
{rperera,pnand,gklette}@aut.ac.nz

Abstract. DBpedia encodes massive amounts of open domain knowledge and is growing by accumulating more triples at the same rate as Wikipedia. However, the applications often require natural language formulations of these triples to present the information as a natural text. The RealText$_{lex2}$ framework offers a scalable platform to transform these triples to natural language sentences using lexicalization patterns. The framework has evolved from its previous version (RealText$_{lex}$) and is comprised of four lexicalization pattern mining modules which derive patterns from a training triple collection. These patterns can be then applied on the new triples given that they satisfy a defined set of constraints.

1 Introduction

DBpedia has become a central hub for the applications searching for information on the web. Since this information is provided in structured form as triples, the applications require the natural language formulation of these triples. In essence, an application that needs to provide biographies would need to transform a selected set of triples to natural language in order to present it as a natural text. This approach gives more freedom to content owners to concentrate on the actual content rather using naive techniques to retrieve content from another unstructured text resource using summarization or other approaches.

Transforming triple-like meaning representations into natural language is termed as lexicalization - a subtask of Natural Language Generation. RealText$_{lex2}$[1] (refer [1] for RealText$_{lex}$) approach for lexicalization is based on an ensemble architecture comprising of four pattern mining modules. Three of them are based on specially crafted lexicons and the others extract patterns from unstructured text using Open Information Extraction (OpenIE) and make them cohesive, so that they can be generalized. This is a completely different approach compared to the available Linked Data lexicalization platforms; *corpus based approach* [2] which extracts bare typed-dependency paths as patterns, and *LOD-DEF* [3], which substitutes the triple subject and object in a sentence to form a pattern. A definition of lexicalization pattern in our approach is another triple structure

[1] A video demonstration is available at https://vimeo.com/173608664

which *S?* and *O?* expressions denote subject and triple respectively. As an example, a pattern such as $\langle$*S?, was born in, O?* $\rangle_L$ can be used to lexicalize the triple $\langle$*Steve Jobs, birthDate, 1955-02-24*$\rangle_T$.

The rest of the paper provides details on the framework and all features presented herein will be part of the demonstration.

2 Demonstration

For the demonstration we utilize the Java client application. Although it shows a similar interface to the previous version, the application layer has been redeveloped for various improvements. These will be discussed in Section 2.2.

2.1 Datasets

For the purpose of this demonstration we focus on randomly selected seven different ontology classes namely: Office Holder, Educational Institute, Mountain, Basketball Player, Country, City, Actor, etc.

2.2 Workflow

The framework is based on four pattern mining modules to generate lexicalization patterns.

Occupational Metonym Patterns Occupational metonyms are used to identify a person based on his/her occupation. In majority of the cases these represent *-er* nominalized verbs (e.g., director, publisher, designer). DBpedia uses occupational metonyms as predicates in multiple scenarios. If such predicate is used then the triple can be lexicalized using the base verb of the *-er* nominalized verb. We have developed a lexicon of such *-er* nominalized occupational metonyms and associated patterns. For example, for a triple such as $\langle$*Now You See Me, director, Jon M. Chu*$\rangle_T$, we can use the pattern$\langle$*S?, is directed by, O?* $\rangle_L$ which is associated with the occupational metonym "*director*".

Context Free Grammar Patterns Context Free Grammar (CFG) is a two directional grammar formalism which helps to both understand and generate language. This research uses only the $S \equiv NP \leftrightarrow VP \leftrightarrow NP$, CFG rule where S denotes a sentence, NP and VP represent noun phrase and verb phrase respectively. Based on this CFG rule, we define the pattern $\langle$*S?, P?, O?* $\rangle_L$ for all triples which satisfy two constraints. Firstly, the triple predicate should be a verb and secondly the verb should have a $NP \leftrightarrow VP \leftrightarrow NP$ in VerbNet.

Relational Patterns Relational pattern are derived from then unstructured text. We first retrieve triples ($\langle$*subject, predicate, object*$\rangle_T$) from number of entities from different ontology classes. Parallel to this process, we also extract text related to each entity considered. This text is preprocessed to tokenize sentences and resolve co-references. We then extract relations ($\langle arg_1$*, rel,* $arg_2 \rangle_R$) from the preprocessed text using Open Information Extraction (OpenIE). The relations are then aligned with retrieved triples (e.g., a triple subject may align with arg_1 of a relation). The alignment is calculated using Phrasal Overlap Measure (POM) for triple subject and object alignments individually and then multiplied to get the final alignment score. We have experimentally determined that a threshold alignment score of 0.21 limits low ranked inaccurate relational patterns being included in the result.

Furthermore, we noticed that grammatical gender of a triple and object multiplicity of a triple can make a lexicalization pattern more specific. For instance, although the triple $\langle$*Barack Obama, spouse, Michelle Obama*$\rangle_T$ cannot be lexicalized with the pattern $\langle$*S?, is the husband of, O?* $\rangle_L$ which is derived using the triple $\langle$*Michelle Obama, spouse, Barack Obama*$\rangle_T$. Although both triples have a same predicate and subjects belong to the same ontology class, grammatical gender of the subject makes the pattern more specific. Similarly, object multiplicity also needs to be considered as an exception. In this case, a predicate can hold either one or more objects. For example, East River has a triple $\langle$*East River, country, United States*$\rangle_T$ and Nile river has triples: $\langle$*Nile River, country, Egypt*$\rangle_T$, $\langle$*Nile River, country, Burundi*$\rangle_T$, and $\langle$*Nile River, country, Ghana*$\rangle_T$. Although pattern $\langle$*S?, is in, O?* $\rangle_L$ can lexicalize the East river triple, the most suitable pattern for Nile River would be $\langle$*S?, flows through, O?* $\rangle_L$. In these cases, we associate each pattern with either of the above two features.

Property Patterns Property patterns consists of predefined set of lexicalization patterns to transform a known predicates to natural language sentences. Table 1 lists the five predefined patterns with examples.

Table 1. Property patterns with examples

Pattern	Predicate	Resulting lexicalization
$\langle S?'s\ P?, is, O? \rangle_L$	height	$\langle Kobe\ Bryant's\ height, is, 1.98 \rangle_{LR}$
$\langle S?, has, O?\ P? \rangle_L$	championships	$\langle Michael\ Schumacher, has, 7\ championships \rangle_{LR}$
$\langle S?, is, O? \rangle_L$	occupation	$\langle Jennifer\ Lawrence, is, an\ actress \rangle_{LR}$
$\langle P?\ in\ S?, is, O? \rangle_L$	largestCity	$\langle Largest\ city\ in\ Canada, is, Toronto \rangle_{LR}$
$\langle S?, P?, O? \rangle_L$	isPartOf	$\langle Scotland, is\ part\ of, United\ Kingdom \rangle_{LR}$

Pattern Search and Realization The pattern search process associates a lexicalization pattern for a given triple by executing one or more of the aforementioned modules. The modules are prioritized in the same sequential order

as they are explained here and if a matching pattern is found then the rest of the modules are not executed. The framework also carries out further realization to the patterns post these lexicalization modules. For instance, if there is a mismatch of a grammatical gender, then we perform a dependency parsing and automatically correct it. Furthermore, for patterns which describes persons who are not alive in present tense are realized into past tense automatically.

3 Evaluation

We performed two evaluations for the lexicalization framework. The first focused on linguistic accuracy and the second was a human evaluation on 40 random sub-sample to rate both readability and accuracy in a 5-point Likert scale. The framework generated 283 linguistically correct lexicalization patterns for 400 triples yielding 70.75% accuracy rate. The human evaluation showed that more than 70% of the sub-sample is rated between weighted average rating of 4.1 and 5 for both accuracy and readability with 0.866 and 0.807 Cronbach alpha values.

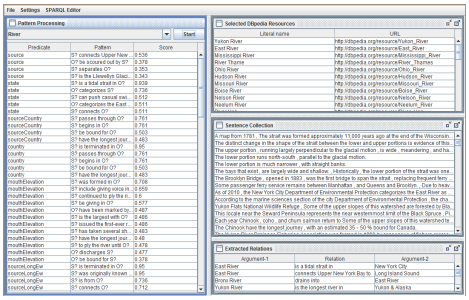

Fig. 1. RealText desktop application. The patterns extracted are shown in the left grid window. The three stacked windows in right show the selected DBpedia resources, candidate sentences, and extracted relations.

4 Conclusion

This paper presented the RealText$_{lex2}$, a framework that uses an ensemble architecture utilizing four separate pattern mining modules. Furthermore, realization is implemented on top of these to increase the pattern's linguistic accuracy. RealText$_{lex2}$ is a part of a larger Natural Language Generation project targeting generating natural language descriptions from the Linked Data cloud. In future, we expect to further enhance the framework with an improved accuracy and a readability level.

References

1. Perera, R., Nand, P.: A multi-strategy approach for lexicalizing linked open data. In: CICLing-2015. (2015) 348–363
2. Walter, S., Unger, C., Cimiano, P.: A corpus-based approach for the induction of ontology lexica. In: NLDB-2013. (2013) 102–113
3. Duma, D., Klein, E.: Generating natural language from linked data: Unsupervised template extraction. In: IWCS-2013. (2013)