Korean Linked Data on the Web: Text to RDF

Martín Rezk[1], Jungyeul Park[2], Yoon Yongun[1], Kyungtae Lim[1], John Larsen[1], YoungGyun Hahm[1], and Key-Sun Choi[1]

[1] Semantic Web Research Centre, KAIST, Daejeon, South Korea
{mrezk,yoon,kyungtaelim, jlarsen, hahmyg,kschoi}@kaist.ac.kr
[2] Les Editions an Amzer Vak, Lannion, France
park@amzer-vak.fr

Abstract. Interlinking data coming from different sources has been a long standing goal [4] aiming to increase reusability, discoverability, and as a result the usefulness of information. Nowadays, Linked Open Data (LOD) tackles this issue in the context of semantic web. However, currently most of the web data is stored in relational databases and published as unstructured text. This triggers the need of *(i)* combining the current semantic technologies with relational databases; *(ii)* processing text integrating several NLP tools, and being able to query the outcome using the standard semantic web query language: SPARQL; and *(iii)* linking the outcome with the LOD cloud. The work presented here shows a solution for the needs listed above in the context of Korean language, but our approach can be adapted to other languages as well.

Keywords: NLP2RDF, Linked Open Data, Korean, RDF, Reasoning

1 Introduction

The Web of Linked Open Data (LOD) is developing rapidly, and with it the number of resources described, which are represented by RDF statements. However, still most of the web data is stored in relational databases and published as unstructured text. Moreover, the number of links between the resources that are already published is low compared with the amount of data published in the LOD cloud—less than 2% at the moment. A way to overcome this gap between the traditional *Web of Documents* and the LOD is to extract facts and links from unstructured text using and chaining the different available NLP tools. In order to allow interoperability between different NLP tools it is desirable to have ontologies that define and establish the vocabulary to be used, and the relation among the different terms in it. However, when this approach is applied to non-Latin languages, such as Korean, there are further issues that need to be solved regarding internationalization (*i18n*) of NLP tools, ontologies, and standards—such as URI vs IRI. Furthermore, it is necessary to minimize the performance gap between relational and RDF data management. This can be done using existing technologies to access and reason with ontological data that is stored in relational databases.

The first task towards linking Korean data with the LOD cloud is to identify which resources and which properties we want to describe. The resources we

are interested in this paper are *morphemes, words (eojeols)* and *sentences* in Korean. We are also interested in modelling linguistic properties such as part-of-speech (POS) information and grammatical roles among others. Our final goal is to chain several Korean NLP tools and be able to efficiently publish the outcome on the web linking Korean text with Korean DBpedia [3]. Summarizing, the issues we need to solve are:

1. **Linguistic Modelling**: We need to model the outcome of the different Korean NLP tools—such as POS–with reference language-independent concepts that allow: *(i)* interoperability with other NLP tools, and *(ii)* conceptual interoperability with other linguistic annotations. To this end, we need ontologies.
2. **Producing RDF triples and Accessing the Data**: We need to align the outcome with the different ontologies, produce the RDF triples, and be able to query and *reason* with this data enhanced by the ontology.
3. **Linking the resources with the LOD cloud**: The final critical step is to link these resources with existing elements in Korean DBpedia.

Our main focus in this paper is to show how we have solved Items 1 and 2 and our first approach towards solving Item 3. In the current work we are linking Korean entities with Wikipedia, and in a follow-up paper we will extend this work to link these entities with the most specific resource in the LOD cloud exploiting the linguistic information provided by the NLP tools. We show preliminary results evaluating this first attempt tackling Item 3.

The contribution of this work is many-fold: *(i)* Ontologies for Korean linguistic annotations; *(ii)* i18n of the NLP Interchange Format; *(iii)* An application (available online) that allows processing Korean text, producing RDF triples, and efficiently querying and *reasoning* with the outcome; *(iv)* Links connecting Korean entities with Korean Wikipedia and preliminary results evaluating this approach. Observe that the method applied here to connect entities with Wikipedia can be applied also to connect them to DBpedia. However, currently Korean DBpedia is under migration tasks.

2 Formats and Ontologies for Korean Annotations

To allow interoperability between different NLP tools, the outcome of these tools must be modelled with formal conceptual descriptions and linked with language-independent reference concepts. To this end, we need ontologies defining these concepts, and specifying the relation among them. Since we focus on describing the linguistic properties of eojeols and Korean sentences; the first step was to identify and model these Strings and then the Sejong tagset—containing POS tags for Korean—and grammatical roles into an OWL ontology (Sejong Ontology) for linguistic annotations.

To describe the resources, that is, eojeols and sentences, we relay on the NLP Interchange Format (NIF) [2]. NIF is an RDF/OWL-based format that can represent Strings as RDF resources. NIF relies on a Linked Data enabled URI scheme and defines two ontologies (String and SSO ontologies) that do not need further modifications to be applied to Korean text. The NIF format is used to *(i)*

standardize the input/output of the different tools to ease to connection among them, and to *(ii)* uniquely identify (parts of) text, entities and relationships. Further details can be found in [2]. To identify the Strings in a text, NIF provides two URI schemes: The *offset* and *context-hash* based schemes. We opt in our application for the latter one since it has several advantages regarding stability of the URI. The Hash-based URIs have five components: *(i)* The word "hash"; *(ii)* the *context* length—that is, a predefined number of characters surrounding the String to left and right; *(iii)* the overall length of the String; *(iv)* a 32 character md5 hash created of the String and the context; and *(v)* a human readable part consisting of the first 20 characters of the referenced string.

Apart for Item *(i)*, this specification cannot be applied straightforwardly to Korean. An eojeol is composed of several Hangul syllables. One syllable is composed of two to four Hangul alphabet symbols. , for instance "금" is one syllable composed of the symbols: ㄱ, ㅡ and ㅁ. Since not every combination of Hangul alphabet symbols form a syllable, it is desirable to keep the syllables atomic and make one Hangul syllable correspond to one character in Items *(iii)-(v)*. In addition, URIs specification do not support Hangul. Korean DBpedia solve this problem by using the percent-encoding of the Korean Strings. However, such encoding is not readable by humans. Thus, we propose to extend the NIF standard to support Hangul alphabet symbols (i18n) and use syllables instead of characters to define the context in the case of Korean Language. In our prototype we use and support both: percent encoding and Hangul alphabet symbols. All the issues mentioned above also increase the difficulty of linking Korean entities with the LOD cloud. This will be explained in Section 4.

To model the linguistic properties of eojeols and sentences, we categorized the Sejong tagset into twenty one tag classes for linguistic annotation—such as ProperNoun, CaseMarker, Determiner—together with their respective hierarchy. In particular, we carefully defined case markers and verbal endings—present in Korean and other non-Latin languages but not in English—in the class "Particle" where significant information concerning syntactic structure is expressed. Furthermore, we added classes such as "LikelyNoun" for particular tags (c.f. Figure 1) which, to the best of our knowledge, do not exist in English tagsets, such as the Penn tagset. Once the Sejong ontology was well-defined, we used the *Ontologies of Linguistic Annotation* (OLiA) [1] to link the ontological concepts from the Sejong ontology with language-independent reference concepts. The OLiA consist of three different ontologies:

1. The OLiA reference model: specifies the common terminology that different annotation schemes can refer to.
2. The OLiA annotation model[3]:formalizes the annotation scheme and the tagsets. In our case, this is the Sejong ontology.
3. The OLiA linking model: defines the inclusion relationships between concepts and properties in the Annotation Model and the Reference Model.

In Figure 1 we show the correspondence between Sejong tags and concepts and the concepts in the OLiA reference model.

[3] The annotation model might consists of several ontologies.

Tag		Sejong	OLiA
superclass		LinguisticAnnotation/Tag/	LinguisticConcept/MorphosyntacticCategory/
MA	MA	Adverb	Adverb
	MAJ	Adverb/ConjunctiveAdverb	Adverb and Conjunction/CoordinatingConjunction
	MAG	Adverb/GeneralAdverb	Adverb
SN, XN		CardinalNumber	Quantifier/Numeral
MM		Determiner	PronounOrDeterminer/Determiner
SH, SL		ForeignWord	Residual/Foreign
IC		Interjection	Interjection
XR		Noun/BaseMorpheme	Noun/CommonNoun
NN	NN	Noun	Noun
	NNB, NNG	Noun/CommonNoun	Noun/CommonNoun
	NNP	Noun/ProperNoun	Noun/ProperNoun
	NA, NF	LikelyNoun	Noun
NP		Pronoun	PronounOrDeterminer/Pronoun
SE, SF, SO, SP, SS		Symbol	Punctuation
V	NV, V	Verb	Verb
	VA	Verb/Adjective	Verb and Adjective/PredicativeAdjective
	VX	Verb/AuxiliaryPredicate	Verb/AuxiliaryVerb
	VC, VCN, VCP	Verb/Copula	Verb/FiniteVerb
	VV	Verb/VerbalPredicate	Verb
E, JK, XP, XS	E, JK, XP, XS	Particle	MorphologicalCategory/morpheme/
	JC, JX	Particle/AuxiliaryPostposition	MorphologicalCategory/morpheme/MorphologicalParticle
	JKB, JKC, JKG, JKO, JKQ, JKS, JKV	Particle/CaseMarker	MorphologicalCategory/morpheme/MorphologicalParticle
	XPN	Particle/Prefix	MorphologicalCategory/morpheme/prefix
	XSA, XSN, XSV	Particle/Suffix	MorphologicalCategory/morpheme/suffix
	EC, EF, EP, ETM, ETN	Particle/VerbalEnding	MorphologicalCategory/morpheme/suffix

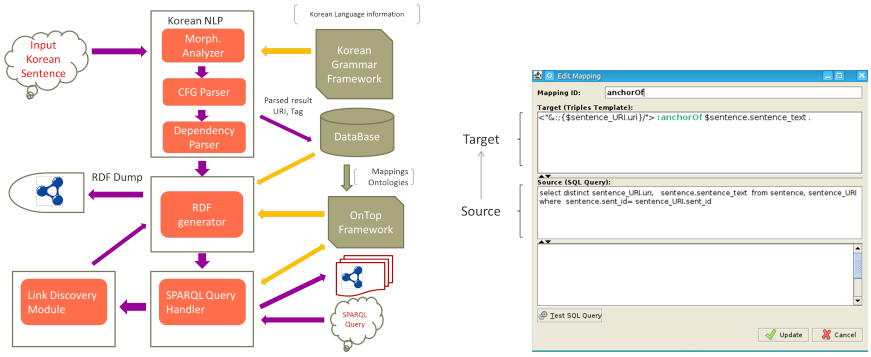
Fig. 1: Correspondence between Sejong tags and the concepts in OLiA

3 Implementing NLP2RDF

Our prototype[4] (Figure 2(a)) takes as input a Korean sentence, runs a number of Korean NLP tools, and displays the result. The outcome of these NLP tools is stored in a relational database (DB). The DB might not be needed now since we only parse one sentence at the time, but in the future we will need it for parsing large amounts of data simultaneously. To make this data available for other NLP tools—that takes an NIF input—or by any Semantic Web application; a user must follow an *ETL*-like process, that is: (E)xtract the data from the sources, (T)ransform it into RDF triples, and (L)oad it into a query answering system [5]. However, as it is, this process has several drawbacks such as generating duplicated data—the parsed sentences and words are in the RDF triple store and in the relational database—decreasing the performance of the system and introducing the problem of synchronizing the DB and the triple store in each update. To avoid these problems our system relies on the Ontology Based Data Access (OBDA) model from the OnTop framework [5]. Our OBDA model is composed by *(i)* the relational DB definition (database, usr, pswd); and *(ii)* a set of mappings representing the relationship between our relational database and the NIF and OLiA ontologies. Figure 2(b) illustrates these concepts.

The `Target` states a class or a property in our ontologies to be defined—such as the NIF property *anchorOf*—and `Source` is a SQL query defining the members/domain-range of such class/property. For instance, in the case of the *anchorOf* property, we need to join the tables containing the IRIs (for the domain) of the word/sentences and the ones containing the String itself (for the range). In this way, we do not need to materialize the set of triples to answer SPARQL queries, although we can also do this materialization if it is needed. It is worth noticing that when we materialize the RDF triples describing the

[4] `http://semanticweb.kaist.ac.kr/nlp2rdf`



(a) Prototype (b) Mapping

Fig. 2

Strings, we only give the triples with the most specific superclass, thus our approach is more efficient space-wise. The rest of the triples can be obtained using **reasoning** and SPARQL queries. For instance, if x has been tagged as a Common Noun, and the ontology states that every Common Noun is a Noun, we do not provide two triples stating that z has $rdf:type$ Noun and CommonNoun, but only one, that is $<$ `z rdf:type :CommonNoun` $>$. However, if we query:

Select ?x where {?x rdf:type Noun}

onTop will use our mappings to reason, and rewrite the SPARQL query into a SQL query in such a way that z will be in the answer. Since SQL is used, we can profit from all the existing optimizations for these tools, and then answer back RDF triples. This closes the *performance gap* between relational and RDF data management. An important feature of this approach in the context in Linked Data, is that our system is aware of the provenance of the data and it also keeps structural information of the data that is lost if we triplify the DB.

To the best of our knowledge there are no other approaches implementing NLP2RDF for Korean, however, we are aware of similar implementations of NLP2RDF for English, for instance, the StandfordCore NIF wrapper.[5] Although the StandfordCore NIF wrapper and our application are similar in nature, the StandfordCore NIF wrapper cannot answer SPARQL queries nor reason as we do. As a consequence, they always produce all the RDF triples that can be derived from the data and the ontologies. It is worth noticing that most of the RDF stores, do not provide reasoning features for query-answering.

4 Towards Linking Korean LRs with the LOD cloud

In this section we tackle the problem of creating links between the entities discovered by a given NLP tool (which output is in NIF) and Korean Wikipedia. This approach can be adapted straightforwardly to link resources with DBpedia; however, at the moment Koeran DBpedia is under migration tasks and it is unstable. Our final goal is to link these resources (words *and* sentences) with the *most* specific DBpedia resource.

As many well known approaches for Link Discovery, such as LIMES[6], we rely on string-based metrics to measure the similarity between entities. Our

[5] `http://nlp2rdf.org/implementations/stanford-corenlp`
[6] `http://aksw.org/projects/limes`

approach first accesses the ontological data using SPARQL and obtains all the nouns. Observe that since the vocabulary is given by the OLiA reference model, it is language and Tagset independent; and moreover, since we allow reasoning we just query the super-class Noun without worrying about the substructure below it. IRIs obtained can represent simple or compound entities. Then using the NIF data property *anchorOf* we get the Strings referenced by those IRIs and check if there is a Korean Wikipedia article which title has Levenshtein distance equal zero (that is, exact matching) with such string. If such article exists, we create the link.

We manually evaluate our approach using sentences picked randomly from news articles. This table shows that our parser tagged more nouns than there actually were originally in each sentence. However with the simple linking method we could link a large portion of the tagged entities with the Korean Wikipedia. Further details are available online[7].

# of Sentences	# of Nouns	# of Tagged Nouns	# of Nouns Linked correctly
16	164	191	119

5 Conclusions and Future Work

In this paper we have presented a solution to publish Korean Strings on the web and a first approach towards linking these resources with the LOD. In order to process Korean text and to produce an RDF output that can be re-used by other NLP tools, we provided ontologies for Korean linguistic annotations, and we suggested an internationalization of the URI scheme of the NLP Interchange Format. We presented a prototype (available online) that allows processing Korean text, producing RDF triples, efficiently querying and *reasoning* with the outcome, and connecting Korean entities with Korean Wikipedia. In addition, we provide preliminary results evaluating this first approach.

Acknowledgments. We thank the anonymous reviewers for useful feedback. This research was supported by the Industrial Technology International Cooperation Program (FT-1102, Creating Knowledge out of Interlinked Data) of MKE/KIAT.

References

1. Christian Chiarcos. Ontologies of linguistic annotation: Survey and perspectives. In *Proceedings of LREC'12*, Istanbul, Turkey, may 2012.
2. Sebastian Hellmann, Jens Lehmann, and Sören Auer. Towards an ontology for representing strings. In *Proceedings of the EKAW 2012*, LNAI. Springer, 2012.
3. E.-K. Kim, Matthias Weidl, K.-S. Choi, and Sören Auer. Towards a Korean dbpedia and an approach for complementing the Korean wikipedia. In *OKCon 2010*, 2010.
4. Mary E. S. Loomis. The 78 codasyl database model: a comparison with preceding specifications. In *Proceedings of SIGMOD '80*, New York, NY, USA, 1980.
5. Mariano Rodriguez-Muro and Diego Calvanese. Quest, an OWL 2 QL reasoner for ontology-based data access. In *Proceedings of OWLED 2012*, 2012.

[7] `http://semanticweb.kaist.ac.kr/nlp2rdf/link.pdf`