Content selection as semantic-based ontology exploration

Laura Perez-Beltrachini CNRS/LORIA Nancy (France) laura.perez@loria.fr
Claire Gardent CNRS/LORIA Nancy (France) claire.gardent@loria.fr
Anselme Revuz I.U.T. Blagnac Toulouse (France) anselme.revuz@gmail.com
Saptarashmi Bandyopadhyay IIEST Shibpur (India) saptarashmicse@gmail.com

1 Introduction

Natural Language (NL) based access to information contained in Knowledge Bases (KBs) has been tackled by approaches following different paradigms. One strand of research deals with the task of ontology-based data access and data exploration (Franconi et al., 2010; Franconi et al., 2011). This type of approach relies on two pillar components. The first one is an ontology describing the underlying domain with a set of reasoning based query construction operations. This component guides the lay user in the formulation of a KB query by proposing alternatives for query expansion. The second is a Natural Language Generation (NLG) system to hide the details of the formal query language to the user. Our ultimate goal is the automatic creation of a corpus of KB queries for development and evaluation of NLG systems.

The task we address is the following. Given an ontology $\mathcal{K}$, automatically select from $\mathcal{K}$ descriptions q which yield sensible user queries. The difficulty lies in the fact that ontologies often omit important *disjointness axioms* and adequate *domain* or *range restrictions* (Rector et al., 2004; Poveda-Villalón et al., 2012). For instance, the toy ontology shown in Figure 1 licences the meaningless query in (1). This happens because there is no disjointness axiom between the Song and Rectangular concepts and/or because the domain of the marriedTo relation is not restricted to persons.

(1) *Who are the rectangular songs married to a person?*
Song $\sqcap$ Rectangular $\sqcap$ $\exists$ marriedTo.Person

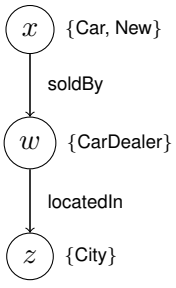
$\top \sqsubseteq \forall$ marriedTo.Person
Person $\sqsubseteq \top$
Song $\sqsubseteq \top$
Rectangular $\sqsubseteq$ Shape

Figure 1: Toy ontology.

In this work, we explore to what extent vector space models can help to improve the coherence of automatically formulated KB queries. These models are learnt from large corpora and provide general shared common semantic knowledge. Such models have been proposed for related tasks. For example, (Freitas et al., 2014) proposes a distributional semantic approach for the exploration of paths in a knowledge graph and (Corman et al., 2015) uses distributional semantics for spotting common sense inconsistencies in large KBs.

Our approach draws on the fact that natural language is used to name elements, i.e. concepts and relations, in ontologies (Mellish and Sun, 2006). Hence, the idea is to exploit lexical semantics to detect incoherent query expansions during the automatic query formulation process. Following ideas from the work in (Kruszewski and Baroni, 2015; Van de Cruys, 2014), our approach uses word vector representations as lexical semantic resources. We train two semantic "compatibility" models, namely DISCOMP and DRCOMP . The first one will model incompatibility between concepts in a candidate query expansion and the second incompatibility between concepts and candidate properties.

2 Query language and operations



Following (cf. (Guagliardo, 2009)) a KB query is a labelled tree where edges are labelled with a relation name and nodes are labelled with a variable and a non-empty set of concept names from the ontology.

The query construction process starts from the initial KB query with a single node. The four operations (cf. (Guagliardo, 2009) for a formal definition of the operations) available for iteratively refining the KB query are: *add* for

the addition of new concepts and relations; *substitution* for replacing a portion of the query with a more general, specific or compatible concept; *deletion* for removing a selected part of the query; and *weaken* for making the query as general as possible. A sequence of query formulation steps illustrating these operations is shown in Figure 2.

I am looking for something. (Initial request)
... for a new car. (Substitution)
... for a new car sold by a car dealer. (Add relation)
... for a new car, a coupé sold by a car dealer. (Add concept)
... for a new car sold by a car dealer. (Deletion)
... for a car sold by a car dealer. (Weaken)

Figure 2: Query formulation sequence.

3 Extracting KB queries

To automatically select queries from a KB, we randomise the application of the *add* and operation. That is, starting from a query tree with one node, the operation is iteratively applied at a randomly selected node up to a maximum number of steps[1]. The *add* operation divides in *add compatible concepts* and *add compatible relations* (cf. (Guagliardo, 2009)). Given a node n labelled with concept s, the first one will add another concept label s' (e.g., Car and New in the example query tree in Section 2), the second will attach a relation and its range (p, o) to the node (e.g., (CarDealer, locatedIn, City)).

The add operation picks up a concept (relation) from a list of candidate concepts (relations) to expand the current query. These candidates are computed using reasoning operations on the query build so far and the underlying ontology. As discussed in Section 1, the lack of axioms in the ontology will enable the inference and the selection of incoherent candidate content such as (Song, Rectangular) and (Song, marriedTo, Person).

To filter out incoherent suggestions made by the add operations we propose the following models[2].

Concept compatibility model (DISCOMP). As explained in (Kruszewski and Baroni, 2015), distributional semantic representations provide models for semantic relatedness and have shown good performance in many lexical semantic tasks (Baroni et al., 2014). While they model semantic relatedness, for instance, *car* and *tyre* are related concepts, they fail to capture the notion of semantic compatibility. That is, there is no thing that can be both a *car* and a *tyre* at the same time. Thus, they propose a Neural Network (NN) model that learns semantic characteristics of concepts classifying them as (in)compatible. We adapt their best performing model, namely 2L-interaction, for our task of detecting whether two ontology concepts (s, s') are incompatible.

Selectional compatibility model (DRCOMP). Selectional constraints concern the semantic type imposed by predicates to the arguments they take. For instance, the predicate *sell* will impose the constraint for its subjects to be, for instance, of type Organisation or Person. Thus, it would be acceptable to say *A car dealer sells new cars* while it would be rare to say *A tyre sells new cars*.

Our idea is to apply the notion of selectional preferences to ontology relations and the concepts they can be combined with. That is, whether a candidate relation p to be attached to a node labelled with concept s, i.e. forming the triple (s, p, o)[3], is a plausible candidate. Along the lines of the work in (Van de Cruys, 2014), we train a NN model to predict (in)compatible subject concept - relation (s, p) pairs[4].

4 Experimental setup

Both models use the best performing word vectors available at `http://clic.cimec.unitn.it/composes/semantic-vectors.html` (Baroni et al., 2014).

DISCOMP dataset. This dataset consists of compatible and incompatible example pairs. We extract them in the following way. We combine a set of manually annotated pairs with a set of automatically extracted ones.

As manually annotated examples, we use the dataset of (Kruszewski and Baroni, 2015) plus additional examples extracted from the results of dif-

[1] A parameter to the random query generation process.

[2] Note that another alternative would be to use the models we introduce to help with the enrichment ontologies.

[3] Note that the concept o taking the object argument place corresponds to the range of the relation. Thus, at this stage, we do not attempt to model relation object concept (in)compatibility.

[4] The architecture of our NN is similar to that proposed by (Van de Cruys, 2014). However, rather than using a ranking loss function, we approximate this by training the network with a hinge loss function over labels (-1,1). Another difference is that our input embedding layer is static and initialised with pre-trained vectors.

ferent runs of the *add* operation which were annotated manually. These provide 7764 examples.

In addition, we automatically extracted compatible and incompatible pairs of concepts from existing ontologies. For incompatible pairs (5273 examples), we extracted definitions of disjoint axioms from 52 ontologies crawled from the web and from YAGO (Suchanek et al., 2008). The compatible pairs (57968 examples) were extracted from YAGO using the class membership of individuals. We assume that if an instance ***a*** is defined as a member of the class ***A*** and of the class ***B*** at the same time then both classes are compatible.

The final dataset contains 71918 instances. We take 80% for training and the rest for testing.

DRCOMP **dataset.** We automatically extract subject-predicate pairs (s, p) from two different sources, namely `nsubj` dependencies from parsed sentences and `domain` restrictions in ontologies.

For the extraction of pairs from text, we use the ukWaCKy corpus (Baroni et al., 2009), we call this subset of pairs *ukWaCKy.SP*, and the Matoll corpus (Walter et al., 2013), call it the *WikiDBP.SP* subset. Both corpora contain dependency parsed sentences. In addition, the Matoll corpus provides annotations linking entities mentioned in the text with DBPedia entities. For the first *SP* dataset, we take the head and dependent participating in `nsubj` dependency relations as training pairs (s, p). For the second *SP* dataset, we use the DBPedia annotations associated to `nsubj` dependents. That is, we create (s, p) pairs where the s component rather than being the head entity mention, it is the DBPedia concept to which this entity belongs to. We do this by using the DBPedia entity annotations present in the corpus. For instance, given the dependency `nsubj(Stan Kenton, winning)`, because *Stan Kenton* is annotated with the DBPedia entity `http://dbpedia.org/resource/Stan_Kenton` and this entity is defined to be of type Person and Artist, among others, we can create (s, p) pairs such as $(person, winning)$ and $(artist, winning)$.

For the pairs based on ontology definitions, we use the 52 ontologies crawled from the web. We call this subset of pairs *KB.SP*.

For training the model, we generate negative instances by corrupting the extracted data. For each (s, p) pair in the dataset we generate an (s', p) pair where s' is not seen occurring with p in the training corpus. The final dataset contains 610522 training instances (30796 from *ukWaCKy.SP*, 571564 from *WikiDBP.SP* and 8162 from *KB.SP*). We take out 600 cases, 300 from *ukWaCKy.SP* and 300 from *KB.SP*, for testing the model on specific text and KB pairs.

5 Evaluation

We separately evaluate the performance of each model in a held out testing set. Table 1 shows the results for the DISCOMP model. Table 2 shows the results obtained when evaluating the DRCOMP model. Both models perform well in the intrinsic evaluation.

Test dataset	Accuracy
(Kruszewski and Baroni, 2015)	0.72
DISCOMP	0.98

Table 1: Results reported by (Kruszewski and Baroni, 2015) and results obtained with the DISCOMP model.

	Test dataset	Accuracy
Emb. + NN	ukWaCKy.SP	0.69
	KB.SP	0.77

Table 2: Results after (Emb.+NN) training with the union of the *ukWaCKy.SP*, *WikiDBP.SP* and *KB.SP* training sets. Note that if we train only with the *ukWaCKy.SP* training set and we evaluate with the *ukWaCKy.SP* testing set we get an accuracy of 0.86 which is similar to the results reported in (Van de Cruys, 2014).

We also asses the performance of the models on the task of meaningful query generation. We run the random query generation process over 5 ontologies of different domains, namely cars, travel, wines, conferences and human disabilities. At each query expansion operation, we apply the models to the sets of candidate concepts or relations. We compare the DISCOMP and DRCOMP models with a baseline cosine similarity (COS) score[5]. For this score we use GloVe (Pennington et al., 2014) word embeddings and simple addition for composing multiword concept and relation names. We use a threshold of 0.3 that was determined empirically[6]. During the query generation process, we registered the candidate sets as well as

[5]For the case of add candidate relations, the COS model checks for semantic relatedness between a subject concept and the relation and between the subject concept and the object concept, i.e. (s,p) and (s,o)

[6]We compare the COS baseline plus a threshold of 0.3

	addRelation		addCompatible	
	COS	DRCOMP	COS	DISCOMP
P	0.51	0.67	0.90	0.88
R	0.30	0.33	0.41	0.85
F	0.38	0.44	0.56	0.87
S	0.79	0.88	0.77	0.46
A	0.59	0.65	0.47	0.78

Table 3: Precision (P), recall (R), F-measure (F), specificity (S) and accuracy (A) results for the DISCOMP , DRCOMP and COS on the add compatible relation (addRelation) and add compatible concept (addCompatible) query expansion operations.

the predictions of the models. In total, we collected 67 candidate sets corresponding to the add compatible relation query extension and 39 to the add compatible operation. The candidate sets were manually annotated with (in)compatibility human judgements. We use these sets as gold standard to compute precision, recall, f-measure and specificity measures on the task of detecting incompatible candidates as well as the accuracy of the models. Figure 3 shows one example for each of the query expansion operations, the annotated candidates and the predictions done by each of the models (only incompatibles are shown).

Table 3 shows the results. Unsurprisingly, given the quite strong similarity threshold used for the COS baseline, we observe that it has good precision at spotting incompatible candidates though quite low recall. In contrast, as shown by the f-measure values the compatibility models seem to achieve a better performance compromise for these measures. We include the specificity measure as an indicative of the ability of the models to avoid false alarms, that is, to avoid predicting a candidate as incompatible when it was not.

6 Conclusions and future work

We applied two compatibility models to get around the lack of disjointness and domain restrictions in ontologies and facilitate the (semi-) automatic generation of a large set of sensible user KB queries. These compatibility models were previously proposed for two semantic tasks. One for term compatibility (Kruszewski and Baroni, 2015) and the other for selectional preference modelling (Van de Cruys, 2014). We automatically created training datasets from several text and knowledge base resources with the intention of providing more adequate training signal for our specific task.

and the COS baseline with 0.5. Setting this threshold is really a trade off between precision and recall. The use of the 0.5 threshold resulted in rejection of most of the candidates including compatible ones.

[Add compatible concept] [Assistant]
[CANDIDATES] [Author:0, SubjectArea:1, Administrator:0, Member_PC:0, Science_Worker:0, Volunteer:0, Scholar:0, Regular:1, Student:0]
[COS] [Member_PC]
[DISCOMP] [SubjectArea, Volunteer, Regular]

[Add relation] [Poster]
[CANDIDATES] [dealsWith:0, writtenBy:0]
[COS] [dealsWith]
[DRCOMP] [ ]

Figure 3: Example of gold standard annotations for the add compatible concept and relation operations and predictions done by the different systems.

As future work, we aim at running a larger task based extrinsic evaluation of these models. We plan to generate a set of KB queries, verbalise them using techniques proposed in (Gardent and Perez-Beltrachini, 2016; Perez-Beltrachini and Gardent, 2016) and ask for human judgements about meaningfulness of the generated queries. In this larger evaluation, we plan to test the models on larger general purpose KBs such as DBPedia.

Further work for improving on the current results could explore the adaptation of the models to specific domain vocabularies and the use of better composition modelling for multiwords concepts and relations.

Acknowledgements

We thank the French National Research Agency for funding the research presented in this paper in the context of the WebNLG project. We would also like to thank Sebastian Walter for kindly providing us with the MATOLL corpus and the volunteer annotator for contributing to the evaluation.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Julien Corman, Laure Vieu, and Nathalie Aussenac-Gilles. 2015. Distributional semantics for ontology verification. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 30–39, Denver, Colorado, June. Association for Computational Linguistics.

Enrico Franconi, Paolo Guagliardo, and Marco Trevisan. 2010. Quelo: a NL-based intelligent query interface. In *Pre-Proceedings of the Second Workshop on Controlled Natural Languages*, volume 622.

Enrico Franconi, Paolo Guagliardo, Marco Trevisan, and Sergio Tessaris. 2011. Quelo: an Ontology-Driven Query Interface. In *Description Logics*.

André Freitas, João Carlos Pereira da Silva, Edward Curry, and Paul Buitelaar. 2014. A distributional semantics approach for selective reasoning on commonsense graph knowledge bases. In *Natural Language Processing and Information Systems*, pages 21–32. Springer.

Claire Gardent and Laura Perez-Beltrachini. 2016. A Statistical, Grammar-Based Approach to Micro-Planning. *Computational Linguistics*.

Paolo Guagliardo. 2009. Theoretical foundations of an ontology-based visual tool for query formulation support. Master's thesis, KRDB Research Centre, Free University of Bozen-Bolzano, October.

Germán Kruszewski and Marco Baroni. 2015. So similar and yet incompatible: Toward automated identification of semantically compatible words. pages 964–969.

Chris Mellish and Xiantang Sun. 2006. The semantic web as a linguistic resource: Opportunities for natural language generation. *Knowledge-Based Systems*, 19(5):298–303.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Laura Perez-Beltrachini and Claire Gardent. 2016. Learning Embeddings to lexicalise RDF Properties. In **SEM 2016: The Fifth Joint Conference on Lexical and Computational Semantics*, Berlin, Germany.

María Poveda-Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. 2012. Validating ontologies with oops! In *International Conference on Knowledge Engineering and Knowledge Management*, pages 267–281. Springer.

Alan Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. 2004. Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In *In Proc. of EKAW 2004*, pages 63–81. Springer.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35.

Sebastian Walter, Christina Unger, and Philipp Cimiano. 2013. A corpus-based approach for the induction of ontology lexica. In *Natural Language Processing and Information Systems*, pages 102–113. Springer.