Identifying Relation-gaps in Ontologies using TOPSIS

Subhashree S and P Sreenivasa Kumar
Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai, India.

Abstract
Though many ontologies have a huge number of classes, one cannot find a good number of object properties connecting the classes in most of the cases. Adding object properties makes an ontology richer and more applicable for tasks such as Question Answering. In this context, determining the pair of classes that are most likely to have object properties between them becomes very important. We address the above question in this paper. We propose a simple yet powerful framework based on the popular multi-criteria decision making algorithm TOPSIS (Technique for Order Preference by Similarity to Ideal Solution), for identifying relation-gaps. In order to identify such relation-gaps (i.e class-pairs) the proposed system utilizes a mix of criteria derived from the ontology itself and from external sources such as Wikipedia text corpora. Our experimental results show that by means of these carefully chosen criteria and their corresponding weights, the proposed system yields promising results with respect to the precision of the relation-gaps identified.

1 Introduction
In this work, we propose a novel and simple approach to identify relation-gaps in an ontology. We say that there exists a relation gap between a pair of classes A and B if an individual of class A can be potentially related to an individual of class B by an object property p but p does not exist in the ontology at present. Informally, we say that p (potentially) connects A and B under the above circumstance. For example, the class-pair (http://schema.org/Language, http://schema.org/Place) of the YAGO4 ontology forms a relation-gap as the two classes could be related by relations such as “is spoken in”, “is the official language of” etc. Note that the focus in this paper is not on discovering the object properties which connect the individuals of a given pair of classes. While there are systems such as OntExt (Mohamed, Jr., and Mitchell 2011) and DARO (Subhashree and Kumar 2020) for the above task, our goal in this work is to identify the pairs of classes which could serve as input to systems such as DARO. Identifying relation-gaps becomes important because, feeding every non-connected class-pair as input to systems such as DARO would be inefficient. This is especially true in the case of large knowledge graphs (KGs) such as YAGO3 which has 488,469 classes but only 77 object properties (Mahdisoltani, Biega, and Suchanek 2015). We employ the TOPSIS method (Technique for Order Preference by Similarity to Ideal Solution) (Hwang and Yoon 1981) which is a standard multi-criteria decision making algorithm, to identify and rank the relation-gaps. We design several criteria derived from the ontology itself and from external sources such as Wikipedia text corpora and provide them as input to the TOPSIS algorithm. Through our experimental evaluation we can see that the proposed approach yields promising results.

2 Related Works
Prophet (Appel and Hruschka Junior 2011) predicts pairs of classes to be connected by object properties, mainly in the NELL KG. Given a pair of classes, Prophet computes a score as the sum of common neighbours (in the KG) of all pairs of instances in the two classes, normalized by the number of instance pairs. The class pairs having a score above 10 are output by Prophet. The main disadvantage of this approach is that if the given ontology does not have many property instances, we cannot expect many new class-pair connections from Prophet. In (Subhashree and Kumar 2020), we have proposed a solution based on word embeddings for the problem of identifying relation-gaps. Word2Vec (Mikolov et al. 2013) has been used for this purpose; those pairs of classes whose labels have high Word2Vec-based similarity are marked as relation-gaps according to our algorithm. The intuition behind this is that word vectors learnt by the Word2Vec algorithm are such that two words which have high number of common neighbouring words have highly similar representations. We have observed two limitations of this Word2Vec-based technique - (1) The class-pair (http://schema.org/House, http://schema.org/Apartment) of the YAGO4 dataset occurs within the top 10 results when we apply the Word2Vec-based method on the YAGO4 dataset. This is because the class labels “House” and “Apartment” have some common neighbouring words. However, one cannot think of many intuitive relations between these two classes (except for a few relations such as “is situated beside”). (2) Word2Vec embeddings are context-independent. For example, the Word2Vec-based technique outputs the class-pair (http://schema.org/Course,
http://schema.org/GolfCourse) of the YAGO4 ontology with a high similarity score because of the common word “Course” in both the class labels. However, the former refers to an academic course while the latter refers to the ground where Golf sport is played. We have observed that the Word2Vec-based approach gives complementary results to that of the existing systems in certain datasets. Hence we utilise it as one of the criteria in our proposed system to detect relation-gaps. Initially, we attempted to build a supervised model to solve the problem of identifying relation-gaps. However, we soon realised that it is close to impossible to get perfect negative samples to train a classifier for this particular problem. Any two objects in the universe can possibly be connected. The level of their connectedness might vary from popularly connected (for example, classes such as “Writer” and “Book” via the relation “authorOf”) to less-commonly connected (for example, classes such as “Cheese” and “Mountain” via the relation “isObtainedFrom”). Thus we realised that we need to build an unsupervised system which can rank the class-pairs based on their level of connectedness.

3 The Proposed Framework

3.1 Criteria for relation-gaps discovery

Given a class-pair (C1,C2) the following criteria are used to decide whether it is a relation-gap or not.

**Criteria (1)-(3):** In (Subhashree and Kumar 2020), we have shown that the common-neighbours measure, Adamic-Adar index, and Word2Vec similarity based measure produce complementary results to each other with good precision values. Hence we adopt these three criteria in our proposed system. We first recall the definition for the neighbour of a class in a knowledge graph: When class C1 is the domain (range) of an object property and class C2 is the range (domain) of that property, classes C1 and C2 are said to be neighbours of each other.

Common-Neighbours (CN): This method captures the number of shared neighbors between both the classes. Let $\Gamma(C_1)$ denote the set of neighbours of a class $C_1$. Then $\text{CN}_{C_1,C_2} = |\Gamma(C_1) \cap \Gamma(C_2)|$.

Adamic-Adar Index (AA): It is defined as

$$\text{AA}_{C_1,C_2} = \sum_{Z \in \Gamma(C_1) \cap \Gamma(C_2)} \frac{1}{\log |\Gamma(Z)|}.$$  

where class $Z$ is a common neighbour for classes $C_1$ and $C_2$.

Word2Vec-based (WV): As described in Section 2, the similarity between the classes is computed using their corresponding Word2Vec embeddings. This similarity is considered to be directly proportional to the rank of the relation-gap they form. We use the word vectors pre-trained on the Google News dataset which has about 100 billion words.

**Criterion (4):** As an additional measure of similarity between class-labels, we make use of their GloVe (Pennington, Socher, and Manning 2014) embeddings. This similarity is considered to be directly proportional to the rank of the relation-gap they form. GloVe focuses on word co-occurrences over the available corpus and its embeddings relate to the probabilities that two words appear together. We have used the 100-dimensional word vectors pre-trained on the combined Wikipedia 2014 + Gigaword 5th Edition corpora which contains 6B tokens and a vocabulary of size 400K.

**Criterion (5):** The similarity between the classes involved is computed using BERT embeddings (Devlin et al. 2019). BERT (Bidirectional Encoder Representations from Transformers) learns the vector representations through Masked Language Modeling (MLM) i.e. it randomly masks words in the sentence and then it tries to predict them. This means that the model looks in both directions and it uses the full context of the sentence in order to predict the masked word. The main advantage of BERT over the other word representations such as Word2Vec and GloVe is that BERT representations are context-dependent while the others are context-independent. For example, the word “bank” would have the same context-independent representation while occurring in the phrases “bank account” and “bank of the river”. On the other hand, context-dependent models generate a representation of each word that is based on the other words in the sentence. We calculate the BERT-similarity between the two classes by considering their descriptions given by the rdfs:comment property in the ontology. We use the BERT embeddings produced by the model all-MiniLM-L6-v2. We use three different ways of computing similarities between the classes because the three types of embeddings used, namely, Word2Vec, GloVe and BERT have been obtained using different external sources.

**Criterion (6):** Popular class-pairs tend to appear in each other’s definition. Hence we check whether the class label of C1 is present in the WordNet definition of class C2’s label. We also check whether the class label of C2 is present in the WordNet definition of class C1’s label. If both these checks turn out to be true we assign a score of 2, if one of them is true we assign a score of 1 and if none of them hold good we assign a score of 0.

3.2 Application of TOPSIS

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) works by choosing the alternative which has the shortest Euclidean distance from the ideal solution and greatest distance from the negative ideal solution. TOPSIS allocates ranks for the alternatives based on the weights and the impacts of the chosen criteria. Weight refers to the weightage of a particular criterion. Impact value of a criterion indicates whether the criterion has a positive or negative impact on the ranking solution. We assign positive impact for all the criteria designed, as all of them work towards increasing the tendency of a particular class-pair to be identified as a relation-gap. The steps involved in the TOPSIS algorithm are:

1. Create a decision matrix $X$ consisting of $m$ class-pairs and $n$ criteria, $X = (x_{ij})_{m \times n}$. The elements of the ma-
2. The matrix $X$ is then normalized to form the matrix $R = (r_{ij})_{m \times n}$ where $r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{n}x_{kj}^2}}$; $i = 1, 2, \ldots m$; $j = 1, 2, \ldots n$.

3. Calculate the weighted normalised decision matrix. An element of this matrix would be $t_{ij} = r_{ij} \cdot w_j$, where $w_j$ takes the weight assigned for the criterion in $j$th column as its value.

4. If column $j$ has been assigned a positive (+) impact, then the best alternative $t_{bj}$ for that column is the maximum value in that column and the worst alternative $t_{wj}$ is the minimum value in that column. If column $j$ has been assigned a negative (-) impact, then the best alternative $t_{bj}$ for that column is the minimum value in that column and the worst alternative $t_{wj}$ is the maximum value in that column.

5. Let $d_{iw}$ be the distance of an ith row from the worst alternatives. Similarly, let $d_{ib}$ be the distance of an ith row from the best alternatives. $d_{iw}$ and $d_{ib}$ are calculated as:

$$d_{iw} = \sqrt{\sum_{j=1}^{n}(t_{ij} - t_{wj})^2}; \quad d_{ib} = \sqrt{\sum_{j=1}^{n}(t_{ij} - t_{bj})^2}$$

6. The TOPSIS score for each row (class-pair in our case) is given by $\frac{d_{ib}}{d_{iw} + d_{iw}}$. Higher the score, better the rank.

## 4 Experiments and Results

The goal of the proposed system is to rank the relation-gaps such that the class-pairs that are highly likely to have an object property linking them are ranked higher. We use two measures to evaluate our system:

1. **Precision@m:** $\text{precision}@m$ is the proportion of the top-$m$ results that are relevant. We fix $m=100$ and perform a manual evaluation of the top 100 results produced by our system. The three (non-author) human evaluators are asked to assess the connectedness of each class-pair ($C1, C2$) on a scale of (1 to 10) based on the following question:

   “Can $C1$ and $C2$ be related by a (at least one) relation?”. We calculate the average of the scores given by the three evaluators and then convert the average score to a binary scale in order to compute the precision: average score greater than 5 is converted to 1 and average score of value less than or equal to 5 is converted to 0.

2. **Correlation:** We calculate the correlation between the scores given by the human evaluators and the TOPSIS scores given by our proposed system. We find the correlation for a sample of 100 class-pairs obtained through stratified random sampling. The evaluators (non-authors) were asked to rate the connectedness of the class-pair on a scale of 1 to 10. Given that the human evaluation scores for a sample are represented as $\{x_1, x_2, \ldots x_m\}$, and the TOPSIS scores are represented as $\{y_1, y_2, \ldots y_m\}$, the Pearson’s Correlation Co-efficient (Pearson 1895) $r_{x,y}$ is given as:

$$r_{x,y} = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2}}$$

where, $m$ is the sample size, and $\bar{x}, \bar{y}$ are the sample means.

The details of all the results obtained along with their manual evaluations and the inter-annotator reliability values can be found in the web link

### 4.1 YAGO4

We pass 31125 class-pairs of YAGO4 schema as input. We empirically assign weights for the criteria included in our TOPSIS algorithm as: Weight for CN=0.1, AA=0.1, WV=0.1, GloVe=0.1, BERT=0.5, WordNet=0.1. From Table 1 we can see that the proposed system achieves a higher precision and a higher positive correlation co-efficient compared to that of the Word2Vec-based system. The top ten class-pairs identified as relation-gaps by the proposed and the existing systems are given in Table 2. Those class-pairs which have been assessed to be less likely to be connected by object properties have been struck out. From Table 2 it can be seen that the Word2Vec-based system assigns a high rank for erroneous pairs such as (http://schema.org/Course, http://schema.org/GolfCourse) because Word2Vec is a context-independent model. The proposed system includes a context-dependent model, namely, BERT, as one of its criteria.

### 4.2 DBpedia

We pass 287478 class-pairs of the DBpedia ontology as input to our proposed system. We empirically assign
weights for the criteria as: CN=0.15, AA=0.1, WV=0.15, GloVe=0.15, BERT=0.15, WordNet=0.15. From Table 3, we can see that the proposed system achieves a higher precision and a positive correlation coefficient value compared to that of the Word2Vec-based system. From Table 4, we can see that the proposed system achieves a higher precision and a positive correlation coefficient value compared to that of the Word2Vec-based system. The main advantage of the proposed system lies in the fact that the weights of the criteria included in the TOPSIS algorithm can be modified to suit the needs of the input ontology. Devising a systematic procedure to determine the weights according to the input ontology needs further exploration and we intend to include it in our future works.

5 Conclusions and future works

In this paper, we have proposed an unsupervised approach for identifying relation-gaps in an ontology. We propose a ranking method based on the popular multi-criteria decision making algorithm TOPSIS. The proposed system has been tested on three huge, cross-domain ontologies and found to be achieving better precision values when compared to the existing system. The main advantage of the proposed system lies in the fact that the weights of the criteria included in the TOPSIS algorithm can be modified to suit the needs of the input ontology. Devising a systematic procedure to determine the weights according to the input ontology needs further exploration and we intend to include it in our future works.

References

[Appel and Hruschka Junior 2011] Appel, A. P., and Hruschka Junior, E. R. 2011. Prophet - a link-predictor to learn new rules on nell. In 2011 IEEE 11th International Conference on Data Mining Workshops, 917–924.

[Devlin et al. 2019] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186.

[Hwang and Yoon 1981] Hwang, C., and Yoon, K. 1981. Multiple Attribute Decision Making: Methods and Applications, volume 186 of Lecture Notes in Economics and Mathematical Systems. Springer.

[Mahdisoltani, Biega, and Suchanek 2015] Mahdisoltani, F.; Biega, J.; and Suchanek, F. M. 2015. YAGO3: A knowledge base from multilingual wikipedias. In CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4–7, 2015.

[Mikolov et al. 2013] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. 3111–3119.

[Mohamed, Jr. and Mitchell 2011] Mohamed, T.; Jr., E. R. H.; and Mitchell, T. M. 2011. Discovering relations between noun categories. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1447–1455. ACL.

[Pearson 1895] Pearson, K. 1895. Note on regression and inheritance in the case of two parents. Proceedings of the Royal Society of London 58:240–242.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), 1532–1543.

[Subhashree and Kumar 2020] Subhashree, S., and Kumar, P. S. 2020. Augmenting linked data ontologies with new object properties. New Gener. Comput. 38(1):125–152.