Similarity of Concepts in Weighted Knowledge Graphs

Yongfan Wang[1[0009−0007−2782−4885]], Ronald R. Yager[2[0000−0003−2938−1858]], and Marek Z. Reformat[1,3[0000−0003−4783−0717]]

[1] University of Alberta, Edmonton, Canada T6G 1H9
{yongfan1, reformat}@ualberta.ca
[2] Iona College, New York 10801, US
yager@panix.com
[3] University of Social Sciences, 90-113 Łódź, Poland

Abstract. Knowledge graphs are recognized as a valuable format for representing data and information. Their ability to represent semantics using different types of relations between the concepts and denoting information at different levels of abstraction creates a demand for algorithms taking advantage of such data format.
In this paper, we propose a method for determining the similarity between concepts in weighted knowledge graphs. The method uses a hierarchical approach to determine the degree of similarity at different levels of 'distance' from the considered graph concepts. The proposed technique employs the T-norm and OWA operator. Similarities between concepts account for edge weights, while OWA aggregates similarities between nodes at different levels of distance from the compared nodes. The method is explained, and its merits are discussed.

Keywords: weighed knowledge graphs · similarity · T-norm · OWA operator

1 Introduction

Understanding and leveraging the complex relationships between concepts is crucial in data semantics. Graphs enable a semantically rich representation of entities and their interconnections and prompt innovative approaches to compare concepts. The graph-based representation of data with concepts as nodes connected by edges denoting relationships (Section 4) offers a framework suitable for addressing concept-related topics. By viewing concepts as nodes, their characteristics, and features are perceived through the edges (relations) that link them to other nodes, Fig. 1. Such a perspective suggests that the similarity between concepts can be determined by examining the nature and context of node relationships.

Motivated by the potential of graph-based representation, this paper introduces a novel approach to utilizing data semantics to ascertain concept similarity within weighted knowledge graphs. Our methodology differs from traditional similarity measures by adopting a feature-based strategy that accounts for *multi-level descriptions* of concepts articulated by direct or indirect links between nodes, Fig. 2. Our approach considers the comparison of concepts in a hierarchical, multi-level manner. Central to our method is applying the Ordered Weighted Averaging (OWA) operator for aggregation similarities determined at different levels of concept descriptions and T-norms as the core aggregation mechanism that includes weights. These tools allow us to integrate diverse features and relationships and capture explicit and implicit attributes describing concepts.

The paper addresses the task of determining concept similarity in weighted knowledge graphs through a feature-based approach. The contributions are twofold. First, we present a comprehensive process for determining the similarity between concepts defined in a weighted knowledge graph. Second, we propose a methodological framework that leverages the OWA operator to aggregate similarities determined based on indirect features of concepts. The process accommodates the multi-layered nature of concept descriptions, i.e., takes into consideration direct and indirect links between nodes. That aspect of our approach is particularly significant, as it enables a more realistic evaluation of concept similarity, moving beyond surface-level comparisons, performed based on the direct links only, to uncover deeper semantic connections.

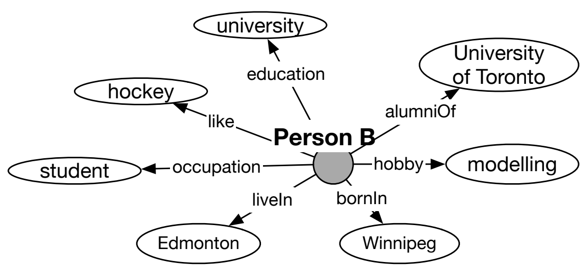

Fig. 1: Graph node = concept definition

2 Related Work

The knowledge graph is a type of graph structure that can store information as nodes and edges, normally the nodes represent entities and the edges represent the relationships between them. The knowledge graph became popular due to its better performance when handling massive amounts of data. Since the development of the Semantic Web, knowledge graphs are often associated with Linked Open Data (LOD) projects, focusing on the connections between concepts and entities [1][2]. Notably, the incorporation of fuzzy sets into weighted knowledge graphs enhances the description of relationships between nodes.

Various methods have been developed to understand the similarity of nodes in the graphs. The Jaccard index [3] stands as a cornerstone for calculating similarities between nodes or sets by capturing shared characteristics. Building on this, adaptations like those by Ravasz et al. [4], which select the smallest sets for comparison, and cosine similarity proposed by Salton [5] has been widely used in citation networks. These methods mainly focus on neighbor nodes and use shared common nodes as the primary criterion for similarity.

To broaden the analysis beyond immediate connections, Symeonidis et al. [6] introduced measures for transitive node similarity, considering indirect connections as well. Around the same time, Tiakas et al. [7] also presented work on employing transitive node similarity to graph node clustering. These studies demonstrate the utility of transitive similarity in tasks such as link prediction and clustering.

However, the discussion around how determining similarities extends to indirectly connected nodes is notably limited. These nodes may share significant commonalities, which should also contribute to the overall similarity. Furthermore, some basic similarity measures may not be suitable for weighted graphs as well. At the same time, these methods are not currently being widely applied to knowledge graphs.

In response to these gaps, our work proposes a novel method for assessing node similarity in weighted knowledge graphs, accounting for both direct and indirect connections between nodes. This method acknowledges shared attributes and similar sub-graphs, thus enriching the calculation of overall similarity by incorporating indirect contributions from these sub-structures.

3 Tools for Comparison Process

Techniques that are components of our method are presented below. We briefly describe each of them.

3.1 Tversky Index

The Tversky index was first proposed by Tversky [8] in 1977. It is a similarity measure that extends and generalizes the concept of the Jaccard index and the Sørensen–Dice coefficient [9] [10], which provides a flexible framework for calculating the similarity and diversity of sample sets. It can be also used in graph theory to determine the extent to which two nodes are similar. Unlike symmetric measures, the Tversky index introduces asymmetry through two parameters, α and β which control the relative importance of the common features versus the distinctive features of the sets being compared. For sets A and B the Tversky index is a number between 0 and 1 given by:

$$T(A,B) = \frac{\mid A \cap B \mid}{\mid A \cap B \mid + \alpha \mid A - B \mid + \beta \mid B - A \mid} \tag{1}$$

Here, $\mid A \cap B \mid$ represents the size of the intersection between sets A and B, i.e. the count of shared elements in both sets. $\mid A - B \mid$ and $\mid B - A \mid$

represent the sizes of the differences between the sets. The parameters α and β can be adjusted to fit the asymmetric measures in calculating similarity. When $\alpha = \beta = 1$, the Tversky index becomes equivalent to the Jaccard index, which is defined as the size of the intersection divided by the size of the union of the sample sets: $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

Our proposed algorithm drew inspiration from both the Jaccard index and the Tversky index to implement both symmetric and asymmetric measures. In our case, the sets we deal with are collections of features that are elements of concept definitions.

3.2 InHyp-OWA

The process of aggregation is uniting various numerical inputs into a single representative value. Our proposed algorithm is specifically designed to calculate the similarity between two root nodes in the weighted knowledge graph. This algorithm aims at both the explicit and implicit relation between nodes, indicating that some sub-nodes of the root nodes may exhibit similar structures and relationships contributing to the similarity measures. As we delve deeper into the multi-level relations, it becomes necessary to find a method for aggregating the similarities from each level. Hence, we employ the ordered weighted averaging (OWA) operator as the aggregation operator. Initially introduced by Yager in 1988 [11], the OWA operator has since found widespread application in the field of fuzzy sets, leading to the development of numerous families of OWA designed for specific contexts and applications.

In our proposed algorithm, as the levels deepen, their contribution will be less to the global similarity. Therefore we need a group of monotonically decreasing weights for aggregation. In 2019, Kishor A. et al. [12] proposed InHyp-OWA, a new family of OWA that generates unique weight vectors W. This group of OWA operators has the feature of generating weight vectors with strictly monotonic components, which aligns well with our requirements. The weight function of InHyp-OWA and how orness is related to the parameter V are presented below

$$w_{i,n}(V) = w_i = \binom{V+n-1-i}{n-i} \Big/ \binom{V+n-1}{n-1} \tag{2}$$

$$orness(W) = \frac{V}{V+1} \tag{3}$$

where n is the finite number of attributes to be aggregated, and V is the parameter that determines orness – a measure of InHyp-OWA being between ‘and’ and ’or’. Orness indicates to what degree the InHyp-OWA behaves as an OR operator and resembles a maximization operation. It is distinctive for InHyp-OWA that the orness value is a function of the parameter V, not a number of aggregated attributes/inputs. Some example weight vectors W’s generated by InHyp-OWA are shown in Table 1.

Table 1: Weights Generated by InHyp-OWA

n	V	weight vector
2	2	(0.6667,0.3333)
	3	(0.7500,0.2500)
	4	(0.8000,0.2000)
	5	(0.8333,0.1667)
3	2	(0.5000,0.3333,0.1667)
	3	(0.6000,0.3000,0.1000)
	4	(0.6667,0.2667,0.0667)
	5	(0.7143,0.2381,0.0476)
4	2	(0.4000,0.3000,0.2000,0.1000)
	3	(0.5000,0.3000,0.1500,0.0500)
	4	(0.5714,0.2857,0.1143,0.0286)
	5	(0.6250,0.2679,0.0893,0.0179)

3.3 T-norm

Understanding the similarity between two nodes in an unweighted graph is straightforward. This similarity is defined by the shared relations and tail nodes, representing the intersection of the two nodes being compared. However, when the graph becomes weighted, this method of intersection may not be accurate enough to describe their relationship adequately. Therefore, the field of fuzzy logic and fuzzy set theory offers a refined approach. The $\mathbb{T}$-norm (Triangular Norm) as a fundamental concept in the field, is a kind of binary operation used in the framework of probabilistic metric spaces and fuzzy logic [13]. Any $\mathbb{T}$-norm can be regarded as natural extensions to realize the set intersection operation of fuzzy sets [14]. Meanwhile, the differences each $\mathbb{T}$-norm brings to the results provide a means to balance precision and inclusiveness. Therefore, here, we use the $\mathbb{T}$-norm as the numerator in similarity calculations to substitute the original set intersection operation. Three common $\mathbb{T}$-norms are selected for result comparison:

$$\mathbb{T}_M(a,b) = min(a,b) \tag{4}$$

$$\mathbb{T}_P(a,b) = a \cdot b \tag{5}$$

$$\mathbb{T}_L(a,b) = max(a+b-1,0) \tag{6}$$

Equations (4), (5), and (6) correspond to the Minimum $\mathbb{T}$-norm, Product $\mathbb{T}$-norm, and Łukasiewicz $\mathbb{T}$-norm, respectively. We will discuss the effects of different t-norms later in the case study.

4 Comparison of Concepts

Most existing similarity measurements designed for comparing graph nodes focus on direct relationships between nodes. However, in the case of knowledge graphs, nodes can be interpreted as representations/definitions of concepts. If a single node A is considered, all its connections to other nodes are treated as features. For example, looking at Fig. 1, it can be said that **Person B** has such features as: is a student, born in Winnipeg, lives in Edmonton, university education, alum of UofT, has modeling as a hobby, and likes hockey.

Further, when nodes in a graph are perceived as definitions of concepts, they can be indirectly connected via other nodes that, further, can be connected directly or indirectly. Let us look at two concepts **Person A** and **Person B**, Fig. 2. They are connected directly via *university*, *hockey*, and *student* (*level-0*), and indirectly via nodes *Calgary* and *Edmonton* (*level-1*); these on the other hand are connected directly via *Alberta* and *energy* and indirectly via *Flames* and *Oilers* (*level-2*). Such a hierarchical multi-level structure should be considered when determining the similarity between $Person\_A$ and $Person\_B$.

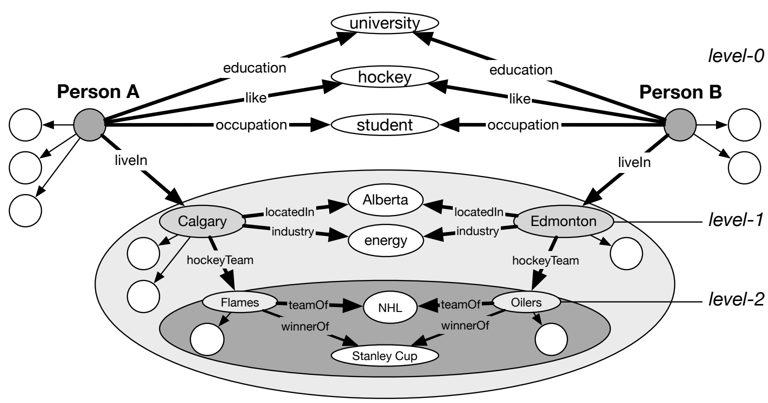

Fig. 2: Direct and indirect connections between two concepts $Person\_A$ and $Person\_B$.

Yet, traditional algorithms hardly consider these indirect connections, leading to inaccurate results. To address this issue, our proposed algorithm compares the nodes and relationships recursively, calculates similarity on a level-by-level basis, and finally aggregates these level-specific similarities by applying an OWA operator.

4.1 Direct Connection – Explicit Similarity Component

In knowledge graphs, concepts/nodes are connected by relations, and each relationship as a triple $\langle head, relation, tail \rangle$. There are four scenarios how two nodes can be connected, Fig. 3.

The explanations and their 'contributions' to the similarity between two nodes are:

(a) *head* nodes are connected by different *relations* to two distinct *tail* nodes; this scenario does not contribute to the node similarity;
(b) *head* nodes are connected to the same *tail* node through different *relations*, the semantic meaning of these relations can vary significantly; for example,

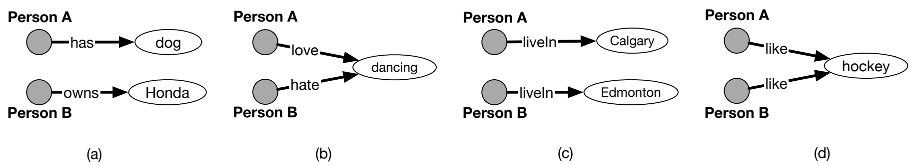

Fig. 3: Four scenarios when comparing two nodes

relations **love** and **hate** are semantically opposite and they should not be considered towards the node similarity;

(c) *head* nodes are connected by the same *relation* to two distinct *tail* nodes; the *tail* nodes might be similar – in such case, they should be considered in determining the similarity between the *head* nodes; the *tail* nodes serve as the initial nodes for recursively calculating implicit similarity, called hereafter a lower level similarity;

(d) *head* nodes are connected via identical *relation* to the same *tail* node, we refer to this scenario as 'direct' similarity.

The description of the proposed method for similarity determination can start with the process of calculating direct similarity. The diagram referring to such a scenario is presented in Fig. 4. Two nodes A_0 and B_0 are connected via a number of relations that are pair-wise, via a connecting node, the same. Each connection, relation, is weighted with α_i^0 on the side of node A_0 and β_i^0 on B_0.

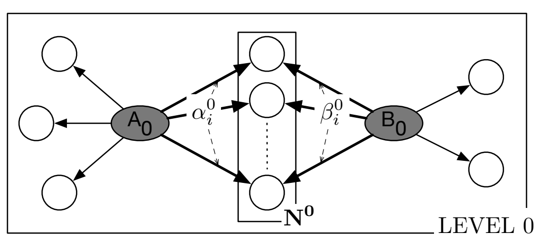

Fig. 4: Direct similarity at level 0

The equation used to calculate the direct similarity is

$$S^0 = \frac{\sum_{i=1}^{N^0} \mathbb{T}(\alpha_i^0, \beta_i^0)}{max(N_{A^0}, N_{B^0})} \tag{7}$$

where N^0 represents a number of common nodes between nodes A_0 and B_0, while N_{A^0} and N_{B^0} are the sums of all weighted relations of node A_0 and B_0, respectively. The superscript 0 indicates the *0-level*.

4.2 Hierarchical Structure – Implicit Similarity Component

Two-level based Similarity As mentioned earlier, nodes *Calgary* and *Edmonton* in Fig. 2, the proposed method takes into consideration 'indirect' similarity that is associated with the nodes that are connected to the considered nodes via the same relation, case c) in Fig. 3

If both *level 0* and *level 1* are considered, the aggregation is fully controlled by the user. They provide a coefficient α that determines a degree of contribution of the *level-1* S^1 similarity towards the total similarity S^0. In this way, the *lower level* similarity (Section 4.1) contributes partially, in a limited way, to the similarity between nodes. The process of calculating the similarity using both levels is represented by the following formula

$$S^0 = \frac{\sum_{i=1}^{N^0} \mathbb{T}(\alpha_i^0, \beta_i^0) + \alpha * \left[\sum_{j=N^0+1}^{N^0+K^0} \mathbb{T}(S_j^1, \mathbb{T}(\alpha_j^0, \beta_j^0)) \right]}{max(N_{A^0}, N_{B^0})} \tag{8}$$

$$S_j^1 = \frac{\sum_{i=1}^{N^1} \mathbb{T}(\alpha_i^1, \beta_i^1)}{max(N_{A^1}, N_{B^1})} \tag{9}$$

here K^0 is a number of pair of nodes $\langle A_j, B_j \rangle$ that are connected to each other via other nodes, box *LEVEL 1* in Fig. 5. As it can be seen, the similarity at the *level 1* (Eq. 9) is calculated in the same way as at the *level 0* (Eq. 7). This lower-level similarity S_j^1 is further aggregated with the weights α_j^0, β_j^0 of connections to this pair using $\mathbb{T}$-norm. After that, it is multiplied by the coefficient α provided by user and then added to the *level 0* similarity.

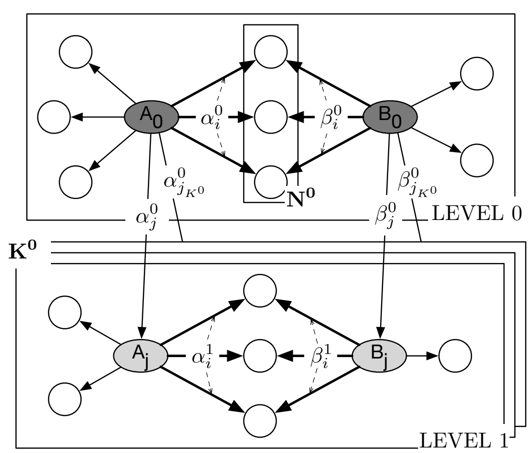

Fig. 5: Indirect similarity at level 1

Multi-level based Similarity If more levels are used to determine similarity, the same approach is recursively repeated. However, this time, the aggregation of similarities from different levels is performed using the InHyp-OWA operator (Section 3.2). It enables a human-like aggregation of similarities when the weights determining the contributions of similarities obtained for various levels are decided via OWA. A general equation for similarity at level Q is

$$S^Q = \frac{\sum_{i=1}^{N^Q} \mathbb{T}(\alpha_i^Q, \beta_i^Q) + w_{Q+1} * \left[\sum_{j=N^Q+1}^{N^Q+K^Q} \mathbb{T}\Big(S_j^{Q+1}, \mathbb{T}(\alpha_j^Q, \beta_j^Q)\Big) \right]}{max(N_{A^Q}, N_{B^Q})} \tag{10}$$

where S_j^{Q+1} is a similarity value calculated for a pair of nodes j at the level $Q+1$, and there are K^Q pairs. Once the similarities obtained for each pair are summed-up, the obtained value represents a contribution from the 'lower' level $Q+1$. That value is multiplied by an OWA weight w_{Q+1}, which are calculated using Eq. 9 for $n = maxDepth$. The $maxDepth$ is a number of levels that the user requires to be included in the calculation of the similarity between the pair of nodes A_0 and B_0. For the last level $Q = maxDepth$, the similarity is calculated by

$$S^{maxDepth} = \frac{\sum_{i=1}^{N^{maxDepth}} \mathbb{T}(\alpha_i^{maxDepth}, \beta_i^{maxDepth})}{max(N_{A^{maxDepth}}, N_{B^{maxDepth}})} \tag{11}$$

Algorithm The Algorithm 1 represents the steps of the proposed method. The main loop (lines 8-12) performs the one-to-one comparison of features. If relations are the same and an object is shared a direct similarity is calculated $dirS$. In the case of the same relations, but different objects (tails) the information about the objects (different) and weights associated with the triples are added to the list $indSC_Npair$ representing nodes that will be compared at the *next level*, and their contributors will be added after multiplication by a weight (provided by users if only two levels are considered, or OWA otherwise). The process of setting up weights – α from a user, or OWA – is performed in lines 13-18. For the OWA, the weights depend on the number of shared objects (tails, line 17) and maximum depth (number of preset levels – $maxDepth$), line 18.

The recursive nature of the algorithm is seen in the second loop by applying Depth-First Search (lines 23-25). The interaction over the list of nodes to compare $indSC_Npair$. Once all similarities between the nodes are determined $indS$, they are aggregated with $dirS$ (line 27).

Lines 19-22 contain simple calculations of the cardinalities of concept features. Since we deal with weighted graphs, the cardinality is a sum of weights.

5 Case Study

The illustration of the proposed methods is shown via the inclusion of a few simple cases. They clarify how the approach works when the calculation of simi-

Algorithm 1: Algorithm: **NodeSim**

```
Global : α, maxDepth;
Input  : node_A, node_B, A.weight, B.weight, level = 0
Output: Sim(node_A, node_B)
if level ≥ maxDepth then
    return 0
triples_A ← node_A_triples;
triples_B ← node_B_triples;
for i in triples_A do
    for j in triples_B do
        if i.relation = j.relation then
            if i.tail = j.tail then
                dirS ← dirS + (𝕋-norm(i.weight, j.weight));
            else
                indSC_Npair.append(i.tail, j.tail, i.weight, j.weight)
if level = 0 then
    if maxDepth ≤ 2 then
        w_1 ← α;                                   /* user entered α */
    else
        v ← dirSC_list.length() ;
        w ← InHypOWA(v, maxDepth);   /* calculation of OWA weights */
cardA = ∑ triples_A.weight;
cardB = ∑ triples_B.weight;
ABsize = max(cardA, cardB);
indS ← 0;
for triplePair in indSC_Npair do
    nA, nB, nA.weight, nB.weight ← triplePair;
    indS ← indS + NodeSim(nA, nB, nA.weight, nB.weight, level + 1);
if level > 0 then
    return 𝕋-norm((dirS+indS)/ABsize, 𝕋-norm(A.weight, B.weight)) · w_level;
if level = 0 then
    return (dirS+indS)/ABsize;
```

larities involves different levels of depth, how similarity depends on a number of features of compared concepts, and what $\mathbb{T}$-norm is used (Section 5.1), as well as when an asymmetry is taken into consideration (Section 5.2).

5.1 Symmetric Approach

The first set of experiments is focused on showing the workings of the approach involving the following cases showing how the results are influenced by: a number of levels – *maxDepth*, a number of common nodes, and a type of $\mathbb{T}$-norm.

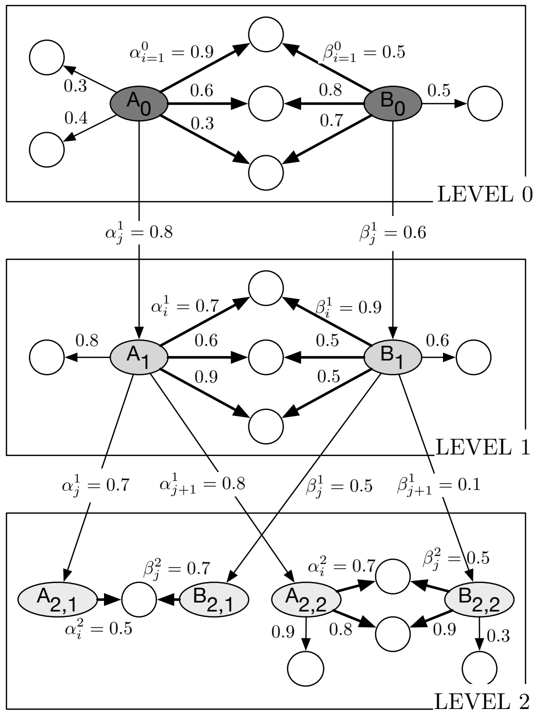

Fig. 6: Case study diagram

The results are shown in Table 2. The upper part of the table includes results when only two levels are considered. In such a situation, the users provide a value that determines the degree of inclusion of the *level 1* similarity to the similarity obtained for the *level 0*. As we can see, the *level 0* similarity is 'fully' included in the calculations (based on Eq. 8 the value of similarity is in $[0, 1]$). The similarity values are for two different numbers of 'common features' shared between two concepts: 3 and 6. The table contains weights which are used to aggregate *level 0* and *level 1* similarities. As expected, adding an additional level increases the values of similarities.

The lower part of the table shows the results when three levels are considered. In this case, an OWA operator is used to aggregate similarities obtained for the levels 0 and 1. We see the values of weights – how they depend on a number of features shared between concepts (V). The more such features concepts have, the smaller portions of similarity values of lower levels are considered.

The values for three different $\mathbb{T}$-norms are presented, Table 2. As we can see, each of them results in different similarity values – *Minimum* $\mathbb{T}$-norm provides the largest values, while *Lukasiewicz* the lowest.

Table 2: Similarity: by number of common nodes, levels, and type of 𝕋-norm

maxDepth	number of features(V)	*level*	(α,OWA) weights	t-norm Minimum	Product	Łukasiewicz
1 (user)	3	0	1	0.4242	0.3455	0.2424
		0&1	α=0.5	0.4815	0.3678	0.2424
	6	0	1	0.5556	0.4296	0.2963
		0&1	α=0.5	0.5905	0.4433	0.2963
2 (OWA)	3	0	1	0.4242	0.3455	0.2424
		0&1	w_1=0.75	0.5101	0.3789	0.2424
		0&1&2	w_2=0.25	0.5177	0.3802	0.2424
	6	0	1	0.5556	0.4296	0.2963
		0&1	w_1=0.857	0.6155	0.4530	0.2963
		0&1&2	w_2=0.143	0.6185	0.4535	0.2963

5.2 Asymmetric Approach

The nature of Tversky's index allows us to consider an asymmetric comparison of concepts. In this case, a similarity value is calculated in the reference to one of the concepts. Let us compare two concepts described by a different number of features. An entity A is described by several features, while only a few describe an entity B. Further, all features of B are a subset of features of the A. In such a case, if the similarity is determined in reference to the B, then the numerator and denominator (Eq. 1) are the same, resulting in the similarity of one. It indicates that if only features of the entity B are considered, the entity B is very similar to the entity A. In the opposite situation, if the features of A are considered, the similarity value is very low (the denominator is much larger than in the numerator).

Illustration of comparison of such very different concepts – size-wise – is shown in Fig. 7. The results of this comparison are shown in Table 3. The values indicates that similarity in reference to the smaller concepts is higher than to the larger ones. In other words, the concepts that have a smaller number of features are more 'similar' to the concepts with a larger number of features. And the concepts that have many features are less 'similar' to concepts with fewer features even in the case that all features of the smaller concepts are a subset of the features of larger concepts.

6 Conclusion

In this paper, we have focused on developing a methodology for evaluating concept similarity within the structure of weighted knowledge graphs. By embracing a graph-based perspective, where concepts are interconnected via edges representing relations between nodes/concepts, we have attempted to provide a technique beyond surface-level comparisons. Our approach, grounded in a feature-based methodology, leverages the richness of hierarchical concept descriptions,

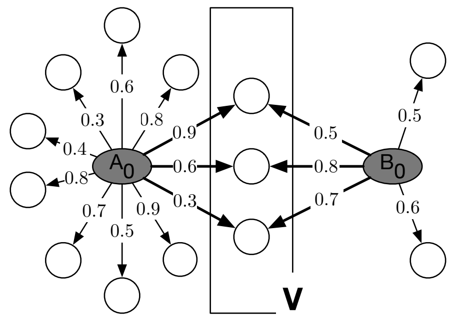

Fig. 7: Comparison of a large concept A_0 with a small one B_0

Table 3: Symmetric vs. Asymmetric Similarities

Similarity	$\lvert N_{A_0}\rvert$	$\lvert N_{B_0}\rvert$	*level*	Tversky Index Denominator $\max(\lvert N_{A_0}\rvert, \lvert N_{B_0}\rvert)$	$\lvert N_{B_0}\rvert$	$\lvert N_{A_0}\rvert$
Symmetric	3.3	3.1	0	0.4242	0.4516	0.4242
	4.5	3.1	1	0.5101	0.5843	0.5101
	0.5 2.4	0.7 1.7	2	0.5177	0.5960	0.5177
Asymmetric	6.8	3.1	0	0.2059	0.4516	0.2059
	4.5	3.1	1	0.2475	0.5843	0.2475
	0.5 2.4	0.7 1.7	2	0.2512	0.5960	0.2512

utilizing the Ordered Weighted Averaging (OWA) operator and T-norms as aggregation mechanisms.

The approach proposed has enabled us to address direct and indirect connections between concepts and offer a more comprehensive framework for similarity assessment. Our work's contributions include developing a detailed process for determining concept similarity and introducing a method that incorporates nested features through the OWA operator.

We intend to extend the proposed approach to develop a feature-based methodology for the comparative analysis of concept graphs. It would enable a better understanding of concepts and reveal relations between clusters of concepts or entire graphs.

References

1. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS), **48**(1-4), 2 (2016)
2. Soylu, A., et al.: Enhancing public procurement in the European Union through constructing and exploiting an integrated knowledge graph. In: The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II 19, pp. 430-446. Springer International Publishing (2020)
3. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat, **37**, 547-579 (1901)
4. Ravasz, E., Somera, A.L., Mongru, D.A., et al.: Hierarchical organization of modularity in metabolic networks. science, **297**(5586), 1551-1555 (2002)
5. Salton, G., McGill, M. J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc (1983)
6. Symeonidis, P., Tiakas, E., Manolopoulos, Y.: Transitive node similarity for link prediction in social networks with positive and negative links. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 183-190 (2010)
7. Tiakas, E., Papadopoulos, A.N., Manolopoulos, Y.: Graph node clustering via transitive node similarity. In: 2010 14th Panhellenic Conference on Informatics, pp. 72-77. IEEE (2010)
8. Tversky, A.: Features of similarity. Psychological review, **84**(4), 327 (1977)
9. Sorensen, T.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biologiske skrifter, **5**, 1-34 (1948)
10. Dice, L. R.: Measures of the amount of ecologic association between species. Ecology, **26**(3), 297-302 (1945)
11. Yager, R. R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on systems, Man, and Cybernetics, **18**(1), 183-190 (1988)
12. Kishor, A., Singh, A. K., Sonam, S., Pal, N. R.: A new family of OWA operators featuring constant orness. IEEE Transactions on Fuzzy Systems, **28**(9), 2263-2269 (2019)
13. Klement, E. P., Mesiar, R., Pap, E.: Triangular norms. Tatra Mountains Mathematical Publications, **13**, 169-193 (1997)
14. Pedrycz, W., Gomide, F.: An introduction to fuzzy sets: analysis and design. MIT press (1998)