TWMS J. Pure Appl. Math., V.12, N.1, Special Issue, 2021, pp.14-28

DEFINITIONS OF CONCEPTS AND IMPRECISION

MAREK Z. REFORMAT[1,2], RONALD R. YAGER[3,4], JESSE X. CHEN[1]

Abstract. Knowledge graphs become an important form of representing data and information. Their intrinsic ability to express semantics via relations enables development of novel methods of processing data and building data models.

In the paper, we propose a methodology for generating definitions of concepts and constructing their hierarchy. It is a fully data-driven process that uses information about multiple entities represented in a form of a knowledge graph. In this work, we state that a concept is defined via relations between the concept and other concepts. We perform a thorough analysis of relations and determine their levels of importance and degrees of their contributions to the definitions. This allows us to include impression reflecting the dependence of the construction process on the context in which it is performed – a limited amount of available data in our case.

We provide details of the proposed approach, and illustrate its performance presenting a case study using a set of facts from dbpedia.org. In the study, we construct a structure of concepts and investigate how importance of relations between them changes when levels of concept abstractions change.

Keywords: knowledge representation, fuzzy logic, logic of vagueness, classification and discrimination, cluster analysis, learning and adaptive systems in artificial intelligence.

AMS Subject Classification: 68T30, 03B52, 62H30, 68T05.

1. Introduction

One of the most important contributions of the Semantic Web concept [1] is the Resource Description Framework (RDF) [13]. This framework is a recommended format of representing data [2]. Its fundamental idea is to represent each piece of data as a triple: <`subject-property-object`>, where the `subject` is an entity being described, the `object` is an entity describing the `subject`, and the `property` is a "connection" between the `subject` and `object`. In other words, the `property-object` is a description of the `subject`. For example, *London is a city* is a triple with *London* as its `subject`, *is_a* its `property`, and *city* its `object`. In general, a `subject` of one triple can be an `object` of another triple, and vice versa. This results in a network of interconnected triples.

The network of triples constitutes an environment suitable for developing new methods for analyzing data, and converting it into more structured information. We imply that this ability is essential to build more semantically oriented data models. Such models would lead to a better understating of new and unknown data, increased inference capabilities, and creation of knowledge.

In this paper, we propose a methodology for building a hierarchy of concepts, their generalization, and determination of imprecision associated with the derived definitions of concepts. Each concept definition is seen as a collection of features that define a given concept. The

[1]University of Alberta, Edmonton, Canada
[2]University of Social Sciences, Poland
[3]King Abdelaziz University, Jeddah, Saudi Arabia
[4]Iona College, New Rochelle, New York, USA
e-mail: reformat@ualberta.ca, yager@panix.com, jesse.chen@ualberta.ca
Manuscript received January 2021.

features, on the other hand, are determined by relations between concepts. A detailed analysis of relations existing between entities of clusters is performed to determine features in a form `concept-relation-concept` . They constitute an essential component of concept definitions. Fuzziness is used to realistically represent relations between multiple concepts. It expresses the variance in importance of relations that can be relevant for concept definitions to a different degree [10, 11].

The overview of the main contribution of the paper, i.e., the concept construction process, is presented in Section 2. The individual steps of the process are:

- Constructing concept prototypes, Section 3. The concept prototypes are identified via clustering RDF-based data. Although RDF-based data is equipped with properties indicating its type and subject, building concepts based on similarity of entities contained in the data provides a number of benefits. This process mimics a data-driven and experience-based learning, leads to construction of an extensional-based hierarchy of concept, and allows to determine degrees of membership of entities to the derived concepts.
- Determining names and degrees of membership of entities to concept prototypes, Section 4. The entities that constitute a concept contribute to its name. A list of common labels that describe a concept is built. At the same time, not all entities equally "fit" a definition of concept. A very simple method is presented to determine a degree to which an entity belongs to a given concept prototype.
- Adjusting strength of connections between concept prototypes, and converting them into concept definitions, Section 5. Entities that belong to a concept are processes from the perspective of their connections to other concepts. In such a way we are able to determine representative and generic connections between concepts.

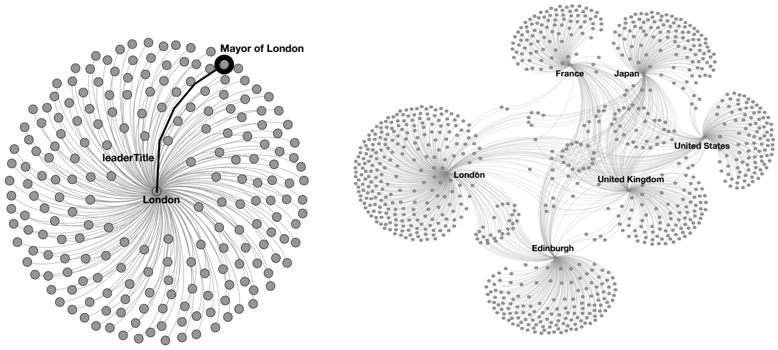
The paper contains a realistic example. More than 50,000 RDF triples have been collected from dbpedia.org and processed. Some of the constructed definitions of concepts are presented in Section 6.

2. Construction of concepts with imprecision: overview

The proposed process of extracting definitions of concepts from data is solely based on processing and analysing RDF descriptions of entities. RDF triples that constitute the descriptions are compared, and levels of similarities between them are determined. They are clustered and the resulting hierarchy of clusters is treated as a structure of concept prototypes. These prototypes are further analyzed and degrees of strength of relations between them are determined. This results in definitions of concepts equipped with imprecision. To summarize, the definitions of concepts are determined by entities that belong to them to a degree, and by relations of different importance existing between.

2.1. Descriptions of entities with RDF.

A single RDF-triple <`subject-property-object`> can be perceived as a feature of an entity identified by the `subject`. In other words, each single triple is a feature of its `subject`. Multiple triples with the same `subject` constitute a description of a given entity. A simple illustration of this is shown in Fig. 1(a). It is a description of London.



(a) (b)

Figure 1. (a) of London with one of its features – relation $\langle London - leaderTitle - MayorOfLondon\rangle$ shown in bold; and of a few entities: London, Edinburgh, France, Japan, United Kingdom and United States all interconnected.

Quite often a `subject` and an `object` of one triple can be involved in multiple other triples, i..e, they can be `objects` or `subjects` of descriptions of other entities. In such a case, multiple entity descriptions can share features. Such interconnected triples constitute a network of interleaving descriptions of entities, Fig. 1(b).

Due to the fact that everything is connected to everything, we can state that numerous entities share features among themselves. In such a case, comparison of entities is equivalent to comparison of their features, i.e., comparison of RDF triples representing the features. This idea is a pivotal aspect of the approach described here for construction of definitions of concepts. It enables categorization, incremental updates, as well as establishing degrees of belonging entities to concepts and a strength of relations between them.

2.2. **RDF Clusters: construction and characterization.** Identification of clusters starts with constructing a similarity matrix. Once a set of triples (RDF descriptions of entities) is obtained, values of similarity are determined for all pairs of RDF descriptions. Such created similarity matrix is an input to an aggregative clustering algorithm. Its result is a hierarchy of clusters (groups of RDF entity descriptions) with the most specific clusters at the bottom, and the most abstract one (one that contains everything) at the top (Section 3).

The next phase is augmenting the obtained clusters which are treated as concept prototypes. Each of them is labeled with a set of features common among all RDF entity descriptions that belong to the same prototype. Elements of the similarity matrix are used to determine the most characteristic – representative – entity for each concept. Degrees of membership of each entity to its concept are also calculated based on the similarity matrix (Section 4).

2.3. **Definitions of concepts and imprecision.** The above-presented processes of clustering and naming is as an initial phase of constructing definitions of concepts with imprecision.

A thorough analysis of concept prototypes, i.e., RDF descriptions of entities that belong to them, is the counter-stone of the process. The principle applied here is based on the fact that a definition of concept is built based on two elements: relations between the concept being defined and other concepts; and the other concepts themselves.

Let us provide a simple example: if we consider a concept of *car* – it is composed of other concepts, such as, *engine, body, wheels* as well as *air conditioning* or *heated steering wheel*. Yet, some of these concepts are essential for a *car*, while some are more like a luxury features, or gadgets. Based on this, we can state that relations between a *car* and its components are of different importance/strength.

To construct definitions of concepts based on this idea, we identify all concepts that 'contribute' to the concept definition as well as all connections between the concept being defined and other concepts. We introduce levels of imprecision associated with degrees of contributions of different concepts to the definition, and strength of relationships between them and the defined concept (Section 5).

3. Clustering of entity descriptions

All interconnected RDF descriptions of entities constitute a graph, and a graph segmentation process could be used to identify groups of highly interconnected – similar – nodes [3, 4, 8, 12]. However, entities – nodes – of the RDF graph play different roles. Some of them are `subjects` of RDF triples (descriptions), and we will call them *defined entities*, while some are just `objects` of RDF triples, we will call them *defining entities*. All nodes which play only the role of *defining entities* should not be involved in the clustering process. Therefore, instead of graph segmentation methods we use an agglomerative hierarchical clustering method that identify nodes of the graph that should be clustered, and the ones that should be excluded from this process.

3.1. **Similarity of RDF entities.** The agglomerative clustering requires a single similarity matrix. The similarity matrix is built with entities as rows and columns. The similarity between entities is calculated using a feature-based similarity measure that resembles the Jaccard's index [7].

In the proposed approach, a similarity value between two *defined entities* is determined by the number of common features. In the case of RDF entity descriptions, it nicely converts into checking how many *defining entities* they shared. The idea is presented in Fig.2. The *defined entities* Edinburgh and London share a number of *defining entities*, and some of these entities are connected to the *defined entities* with the same `property` (black circles in Fig.2).

Figure 2. Similarity of RDF-stars: based on shared objects connected to *the defined entities* with the same properties.

In general, a number of different comparison scenarios can be identified. It depends on interpretation of the term 'entities they share'. The possible scenarios are:

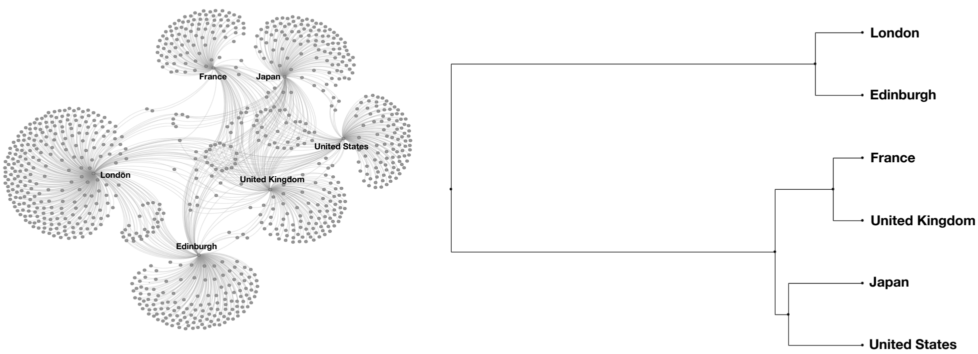
- identical `properties` and identical `objects`;
- identical `properties` and similar `objects`;
- similar `properties` and identical `objects`;
- similar `properties` and similar `objects`;

For details, please see [7]. The similarity assessment process used in the paper follows the first scenario.

3.2. **Similarity matrix and clustering.** A similarity matrix for a set of RDF entity descriptions (*defined entities*) constructed using the similarity evaluation technique presented in the previous subsection is used for hierarchical clustering. The clusters are created via an aggregation process in a bottom-up approach. Two clusters of a lower level are merged to create a cluster at a higher level.

At the beginning each RDF entity description is considered as a one-element cluster. All aggregation decisions are made based on a distance between clusters calculated using an extended Ward's minimum variance measure [9]. This measure takes into account heterogeneity between clusters and homogeneity within clusters. The distances are calculated based on entries from the modified similarity matrix. The modified similarity matrix is de facto a distance matrix created from subtracting the similarity values from a constant equal to the highest similarity value plus epsilon. The two clusters with the smallest distance are merged to become a new cluster. Distances (Ward's measures) between the new cluster and the remaining clusters are calculated. This agglomeration process is repeated until only a single cluster is left.

3.3. **Running example: Clustering.** The described approach to construct categories is illustrated with a simple running example. The data used here is presented in Fig.3. The part (a) is a visualization of six entities – their RDF-descriptions – that constitute an input to the algorithm. They are: London, Edinburgh, United Kingdom, France, Japan, and United States. The part (b) of the figure, shows the clustering results in the form of the dendogram.



(a) (b)

Figure 3. Six entities used for the running example: RDF-stars (a), the dendogram of clustering results (b).

4. From clusters to concepts

4.1. **RDF properties and concept naming.** The clustering algorithm operates on *defined entities*. Once the clusters are defined, we put together all entities. As a result, clusters contain both *defined entities* and *defining entities*.

We 'treat' each cluster as a concept prototype. We label it by a set of names representing features common to all entities that belong to a concept prototype. This is accomplished via taking into account two properties and analyzing all RDF descriptions in a given cluster:

- <`subject`-**dcterm:subject**-`object`>,
- <`subject`-**rdf:type**-`object`>.

We identify all `objects` that are common among all triples with *defined entities* in a single cluster. These `objects` become labels/names of the concept prototype.

We perform this process for all concepts. We start at the top – the most general concept, and go further (deeper) into the hierarchy adding more labels to each concept at the following level. The concepts at the bottom are the most specific, they have the largest number of labels.

4.2. **Concepts, entities, and their membership.** So far, we have treated categories as crisp sets – all RDF entity descriptions fully belong to the concept prototypes. However, when we look closer and inspect values of similarity between members of concepts we see that there are some dissimilarity between entities of the same concept prototype. Therefore, we determine a degree of belonging of a given entity to its concept.

This task starts with identification of the centres of concepts. We extract entries from the similarity matrix that are associated with entities from a given concept, and identify a single entity that has the largest degree of commonality with other entities in the concept. Let C_i be a concept with N entities, and let e_k represents the k-th entity. Its conformance, $conf_{e_k}$, to all other entities from this concept is:

$$conf_{e_k} = \sum_{m=1, m\neq k}^{N-1} sim(e_k, e_m)$$

where $sim(\cdot,\cdot)$ is an appropriate entry from the similarity matrix. Then the centre is:

$$centerID = \arg\max_{n=1 \dots N}(conf_{e_n})$$

We treat this entity as the most representative entity of the concept, and make it its centre. Once the centre is determined, we used its conformance level as a reference and compare it with conformance values of other entities in the concept:

$$\mu_{C_i}(e_k) = \frac{conf_{e_k}}{conf_{e_{centerID}}} \tag{1}$$

In such a way, we are able to determine degrees of membership of entities to the concepts.

4.3. **Running Example: naming and membership degrees.** Now, we name the concept prototypes identified in our running example, Section 3.3, and assign membership values to their entities. The results are shown in Table 1. It contains labels associated with each concept. Please note that the cluster **C1** is labeled with all labels of its predecessors in the hierarchy, i.e., labels of concepts **C5** and **C4**. The values of membership of entities to concepts are given besides entities' names.

Table 1. Running example: naming and membership values for identified concepts.

<table>
<tr><td colspan="3">C5:
Thing, Feature, Place, Populated_Place, Administrative_District, Physical_Entity
Region, YagoGeoEntity, Location_Underspecified
France(0.95), UK(1.00), Japan(0.88), US(0.86), London(0.69), Edinburgh(0.67)</td></tr>
<tr><td colspan="3">C4:
Member_states_of_the_United_Nations, G20_nations
Liberal_democracies, G8_nations
France(1.00), UK(1.00), Japan(0.91), US(0.86)</td></tr>
<tr><td>C1:
Member_states_of_the_EU
Countries_in_Europe
Western_Europe
Member_states_of_NATO
Countries_bordering_the_Atlantic</td><td>C3:
Countries
Bordering
ThePacific
Ocean</td><td>C2:
British_capitals
Capitals_in_Europe
Settlement, City</td></tr>
<tr><td>France(1.00), UK(1.00)</td><td>Japan(1.00), US(1.00)</td><td>London(1.00), Edinburgh(1.00)</td></tr>
</table>

5. Generalization: construction of concept definitions

The proposed methodology for constructing definitions of concepts is considered as a process of generalization. It is driven by analysis of connections between entities that belong to different concept prototypes. Below, we present a description of the methodology using an idealized situation, and then focus on a realistic setting that leads to inclusion of imprecision in the obtained definitions of concepts.

5.1. Concept definitions and relations.

In a nutshell, a process of generalization follows a simple idea: each concept contains a number of entities, and each entity of a given concept, let us say C_i, is linked via relations to entities of other concepts. If multiple entities of the C_i are connected to entities of the same concept C_j via the same property p_V then we imply there is a general connection between these concepts, i.e., there is a triple:

$$\langle concept : C_i - property : p_V - concept : C_j \rangle.$$

Let us analyze such a situation presented in Fig.4.

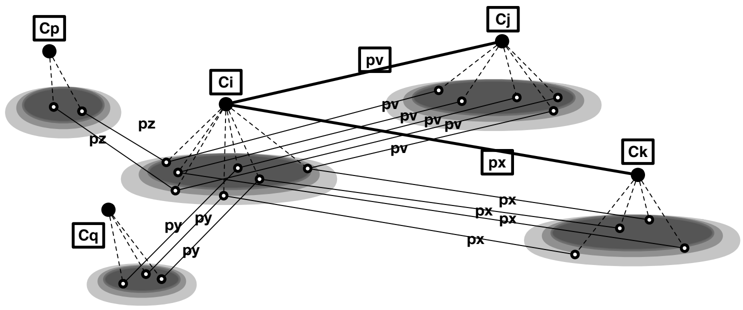


Figure 4. Connections between C_i and other concept: idealized case.

As we can see, the concept C_i contains a number of entities which are connected via properties p_V, p_X, p_Y and p_Z to entities that belong to other concepts. Let us concentrate on two properties p_V and p_X. They connect entities of C_i to entities of C_j and C_k. All connections between C_i and C_j have the property p_V, so we can say that C_j together with p_V is a feature of C_i. Following the same reasoning, we have another feature of C_i that is 'made of' the concept C_k with the property p_X.

If such a process is performed for all properties of entities of the C_i, a set of features of C_i is determined. The final result is an RDF-like representation of the definition of C_i. In other words, we can say that, in our simple example, C_i is defined via triples:

$$\langle C_i, p_V, C_j \rangle,\ \langle C_i, p_x, C_k \rangle,\ \langle C_i, p_z, C_p \rangle,\ \langle C_i, p_y, C_q \rangle.$$

Further, we say that the features of C_i are composed of other concepts together with associated with them properties, Fig.5.

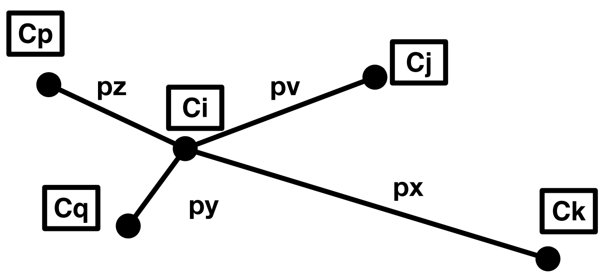


Figure 5. Result of generalization – definition of concept C_i: idealized case.

The presented above scenario is an idealized one. In reality, we deal with two scenarios:

- entities of a given concept are connected with entities of another concept via a number of different properties;
- a given concept is connected with other concepts via the same property.

Such a setting is shown in Fig.6. Entities of C_i are connected to entities of another concept using different properties: C_i is connected with C_j with p_V and p_Y. Additionally, the same property is linked with connections between C_i and other concepts: the property p_X connects C_i also with C_k and C_q.

This observation leads to a premise that features that compose a concept definition should be 'weighted', i.e., we should be aware of the fact that there is some level of imprecision in a statement that a given feature is a part of a definition of concept.

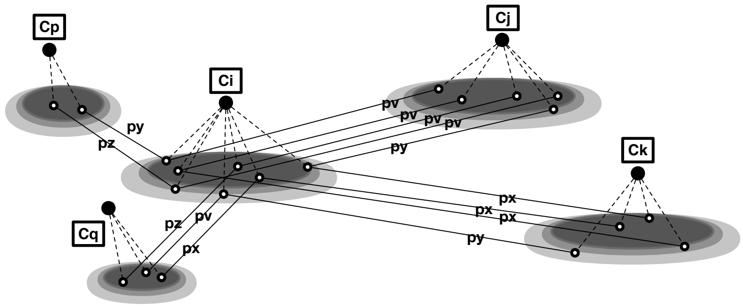


Figure 6. Connections C_i and other concept-clusters: realistic case.

5.2. **Imprecision as fuzziness of relations.** In reality, entities of one concept can be connected – via the same property – to entities that belong to a number of different concepts. Additionally, the clustering process based on similarity matrix can lead to a situation where a number of entities of the considered concept are connected to exactly the same entity of another concept. This is presented in Fig.5 for the property p_V. Please note, the figure shows only connections with the property p_V. Also, it is quite possible that the entities that belong to two different concepts are connected via other properties. As we can see, the property p_V connects the entities of C_i with the entities that belong to three concepts C_j, C_k and C_n. At same time, there is a possibility that these concepts have entities that are not connected to C_i via p_V.

In general, we could say that based on the available data that is used for the clustering process, there are entities of the concept C_i which do have connections to other entities of C_j, C_k and C_n via properties different then p_V, and there are entities of C_j, C_k and C_n that are connected to entities of other concepts via p_V. Based on these two facts we determine a measure of *prominence*. It represents a degree to which a given feature/property contribute to the definition of a concept. The value of *prominence* is calculated using two other measures: *dominance* and *completeness*.

The *dominance* is used to represent popularity of a given property among entities of the concept under consideration. In a nutshell, such a measure would be calculated in the following way, i.e., for two concepts C_i and C_j the *dominance* of p_V is equal to:

$$dominance_{C_i \to C_j}(p_V) = \frac{\#C_i \textit{ entities connected to } C_j \textit{ via } p_V}{\#C_i \textit{ entities}}$$

The value of *dominance* of 1.0 would indicate that all entities of C_i are connected to C_j via the property p_V. However, as we know not all entities 'fully' belong to a given concept – each entity belongs to a concept to a degree, Section 4.2. Therefore, the formula used to calculate the *dominance* is:

$$dominance_{C_i \to C_j}(p_V) = \frac{\sum\limits_{k \in E^p_{i,j}} \mu_{C_i}(entity_k)}{\sum\limits_{m \in E_i} \mu_{C_i}(entity_m)} \tag{2}$$

where $E^p_{i,j}$ is a set of entities of C_i that are connected to entities from C_j via p_V, while E_i is a set of entities of C_i, while $\mu_{C_i}()$ is defined by (1).

The second parameter contributing to the *prominence* of a property is called the *completeness* of p_V:

$$completeness_{C_i \to C_j}(p_V) = \frac{\#\ C_j\ entities\ connected\ to\ C_i\ via\ p_V}{\#\ C_j\ entities}$$

This parameter indicates how many unique entities of the concept C_j are connected to the entities of C_i, or in other words it shows 'exclusiveness' of C_j as a feature of C_i. Its value of 1.0 would mean that all entities of C_j are connected to entities of C_i; yet it does not mean to all entities of C_i. The above formula represents a crisp situation, in reality we deal with membership values of entities. The modified formula has the form:

$$completeness_{C_i \to C_j}(p_V) = \frac{\sum\limits_{h \in E^p_{j,i}} \mu_{C_j}(entity_h)}{\sum\limits_{n \in E_j} \mu_{C_j}(entity_n)} \tag{3}$$

where $E^p_{j,i}$ is a set of entities of C_j to which entities of C_i are connected via p_V, and E_j is a set of entities of C_j.

Finally, the *prominence* of a feature $< ... - p_V - C_j >$ is a product:

$$prominence_{C_i \to C_j}(p_V) = dominance_{C_i \to C_j}(p_V) * completeness_{C_i \to C_j}(p_V)$$

$$prominence_{C_i \to C_j}(p_V) = \frac{\sum\limits_{k \in E^p_{i,j}, h \in E^p_{j,i}} T(\mu_{C_j}(entity_k), \mu_{C_i}(member_h))}{\Big(\sum\limits_{m \in C_i} \mu_{C_i}(entity_m)\Big) * \Big(\sum\limits_{n \in C_j} \mu_{C_j}(entity_n)\Big)} \tag{4}$$

The low values of the *prominence* could be a result of the following situations:

(1) a number of entities of C_i is higher than of C_j;
(2) a number of entities of C_j is higher than of C_i; and
(3) only some entities of C_j are connected with some entities of C_i.

These situations could indicate a limited amount of data used to construct a hierarchy, and a need for more data should be considered. Further, the last one tells that the feature $< ... - p_V - C_j >$ exists only for a few entities of C_i and C_j. This case could lead to a conclusion that this feature should be investigated at different levels of the hierarchy of concepts – possibly at lower levels. These ideas alone, could guide a learning process and refinement of a constructed structure of concepts.

In Fig.7, the feature $< ... - p_V - C_j >$ has the *dominance* of 6/7 and the *completeness* is 3/5. This results in the overall *prominence* of 18/35 (0.514). For $< ... - p_V - C_k >$ the *prominence* is 6/7 times 2/6 (0.286), and for $< ... - p_V - C_n >$ it is 6/7 times 1/4 (0.214).

The presented process of calculations can determine weights of features of the concept C_i. An example of such situation is presented in Fig.8. This situation has to be considered in the

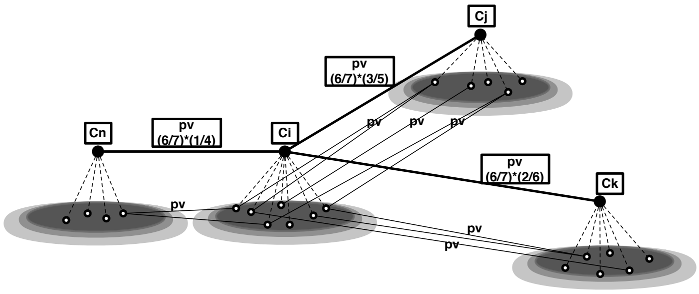


Figure 7. Generalization process: single property p_V.

context of hierarchy of concepts. The fact that the concepts of lower levels are contained in the concepts of higher levels, and the entities of lower level concepts are also entities of higher level concepts leads to a structure of features, Fig.9. We see that the *prominence* levels are calculated starting at the lowest level of the hierarchy and finishing at the level where entities from 'new' concepts are not connected to C_i, i.e., the concepts C_x and C_y are not taken into account.

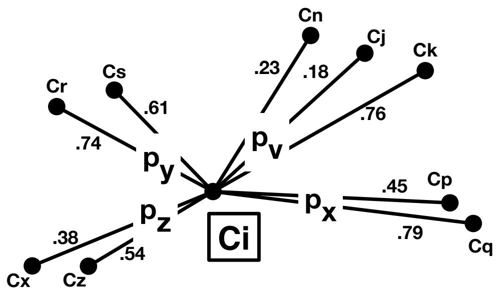


Figure 8. Concept definition of C_i: an example.

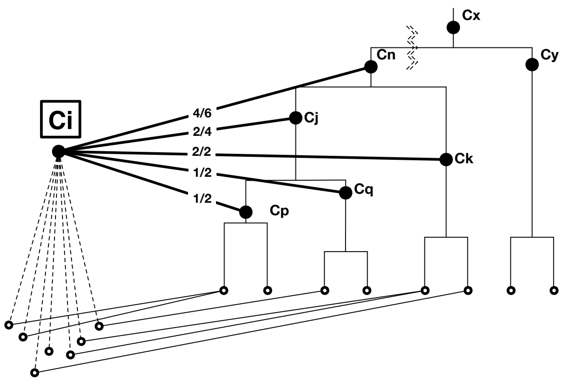


Figure 9. Simple hierarchy of concepts and associated values of *prominence*.

6. Case study

We apply the proposed method to construct definitions of concepts for a relatively large data set. Almost 50k triples have been downloaded from dbpedia.org. The data contains entities from the following categories: people, in particular musicians, actors/directors, writers and scientists; selection of institutions – universities and commercial companies; geographical locations – cities and countries, as well as some examples of human work – movies, games, plays and novels.

After building the similarity matrix we perform a clustering. The dendogram representing the hierarchy of clusters/concepts is shown in Fig.10. As it can be seen a number of concepts has been constructed. The figure includes name of some of them. As it has been indicated this has been done via processing of two properties of entities that belong to each concept: **dcterm:subject** and **rdf:type**, Section 4.1. The generalization process enables us to look at the concepts extracted from RDF data at different levels of granularity. An example of the hierarchal structure could be the concept *Persons* that contains subconcepts *Actor/Director*, *Scientist* and *Musician* to name just a few.

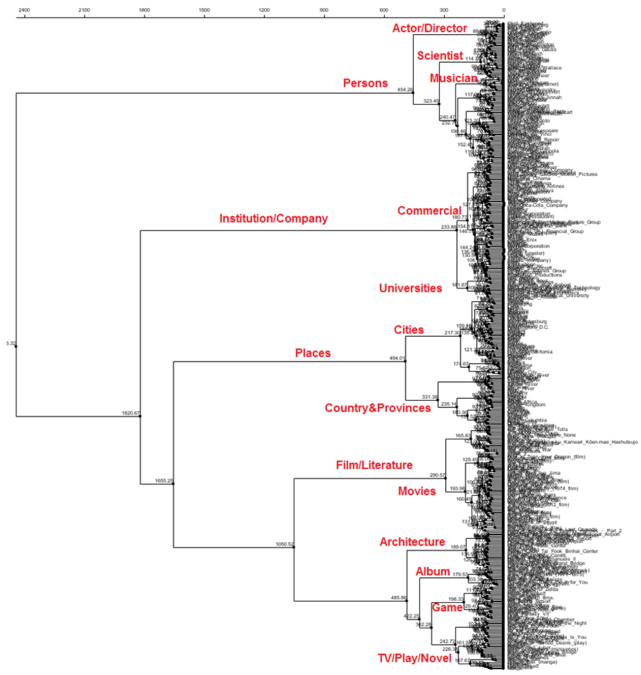


Figure 10. Dendogram of RDF data with indication of identified clusters of entities.

The proposed method not only provides concepts and entities that belong to them, but also gives some insight into relationships between the concepts. We demonstrate this with a simple example based on the obtained structure of concepts, Fig.10.

Let us 'follow' two entities moving up along the concept structure and see how the value of *prominence* of one exemplary relation between the concepts, to which these entities belong, changes. The two entities are *Steven Spielberg*, and *California*. We provide the paths marking the sequences of concepts to which they belong – from very specific ones (at the most right hand side of dendogram) to more abstract ones (at the left hand side of the figure): red for *Steven Spielberg*, and blue for *California*. The relation, or should we say the feature, we investigate is $< ... - birthday - ... >$. We use the term a *source* concept for concepts that contain *Steven Spielberg*, and a *target* source for the concepts with *California*.

To illustrate changes in the values of *prominence*, we look at two scenarios: 1) we fix the concept to which *Steven Spielberg* belongs and see how the *prominence* of *birthday* changes when we 'move' our *target* concept towards more abstract one; and 2) the opposite scenario – we fix the concept to which *California* belongs and consider more and more abstract *source* concepts to which *Steven Spielberg* belongs.

The first scenario is illustrated with the light green arrows in Fig.11. It is important to point out that as the concepts with *California* become more abstract the *prominence* value decreases. This is a result of the diminishing value of *completeness* – there are more entities in the *target* concepts when compared with the entities in the fixed *source* concept.

The second scenario is marked with the dark green, Fig.11. The *prominence* value shows similar behaviour as before, but this time it is the consequence of the fact that concepts with *Steven Spielberg* becomes more general. This time, we see a decrease in the *dominance* measure because more unconnected entities are included in the *source* concept. This results in the diminishing value for the *prominence* value as expected, the dark green arrows.

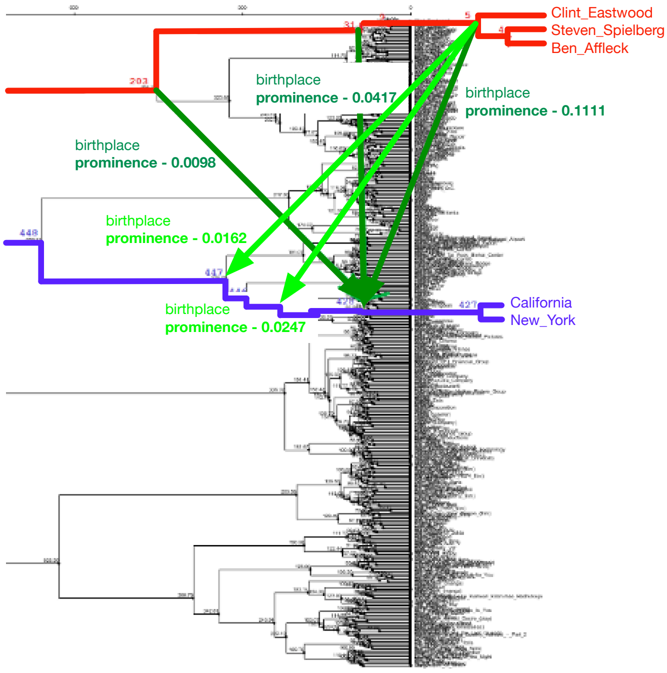


Figure 11. Changes in the *prominence* values of the relation *birthplace*.

In the final example, Fig.12, we focus on another relation – *musicComposer*. Yet, this time we show the values of *prominence* between concepts at different levels of hierarchy. To illustrate different granularity of the concepts and different importance of relations – as the consequence of different context in which this importance has been calculated – we provide a bit more details about relations (marked in orange in the figure) and concepts they connect, Table 2. It shows the labels that indicate naming of concepts, we observe a process of inheritance of labels, together with degrees of strength/importance of the relation *musicComposer*.

Table 2. Running example: naming and membership values for identified concepts.

subject	property	object
Film	**musicComposer** (0.0130)	**Person**
CreativeWork		CausalAgent
Work		PhysicalEntity
Movie		
all above +	**musicComposer** (0.6667)	**all above +**
English-language		Artist
American-movie		Musician
Psychological feature		Composer
		Male-film-score
		Academy-winner
		Grammy-winner

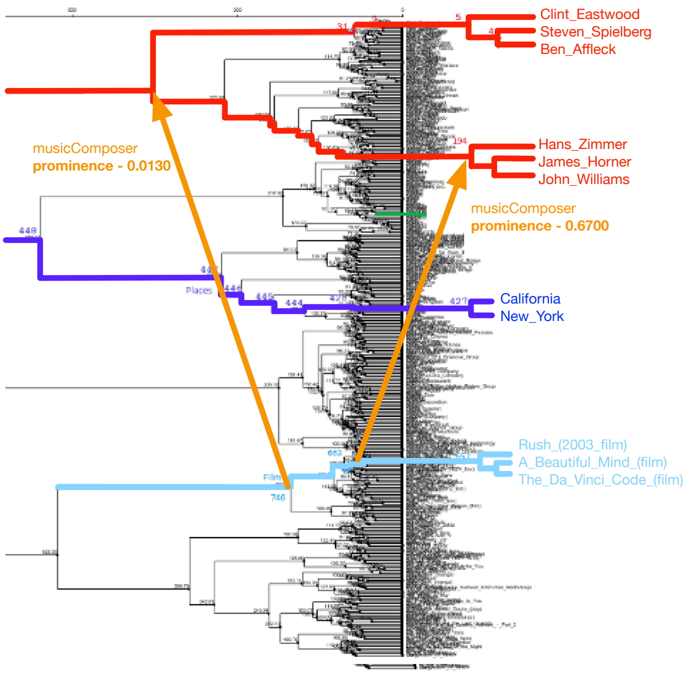


Figure 12. Generalization of concepts and relations.

7. Conclusion

One of the most interesting graph data formats is the Resource Description Framework (RDF). It has been proposed as a part of Semantic Web initiative for representing data and information on the web. Knowledge graphs are characterized by high connectivity – their nodes are linked to each other via different types of relations.

In the paper, we take advantage of a relation-rich structure of knowledge graphs and propose a methodology for constructing a structure of concept definitions. An important aspect of the proposed method is the fact that it is a data-driven process. Once the available data is clustered we analyze the data entities that are instances of the constructed definitions of concepts, and relations between them. A thorough investigation of the relations allows us to determine the degree to which they contribute to the definitions. This allows us to build concepts that are equipped with impression as the consequence of an intrinsic lack of definite agreement what constitutes a given concept, as well as the dependence of definitions on the context in which they are built – available data in our case.

Our experiments indicate that the composition of concept definitions and membership of their instances depend on the considered levels of abstraction. This means that the degrees to which different relations contribute to the definitions are changing.

The next stage focuses on processes of gradual learning of categories based on data that are being collected. The resulting hierarchy of concepts and their definitions can be updated via a continuous inflow of new data.

References

[1] Berners-Lee, T., Hendler, J., and Lassila, O., (2001), The Semantic Web, Sci. Am., pp.29-37.

[2] Christodoulou, K., Paton, N.W., and Fernandes, A.A.A., (2013), Structure Inference for Linked Data Sources using Clustering, EDBT/ICDT Workshops, pp.60-67.

[3] Ferrara, A., Genta, L., and Montanelli, S., (2013), Linked Data Classification: a Feature-based Approach, EDBT/ICDT Workshops, pp.75-82.

[4] Giannini, S., (2013), RDF Data Clustering, Business Information Systems Workshops, Lecture Notes in Business Information Processing, 160, pp.220-231.

[5] Gurrutxaga, I., Arbelaitz, O., Marin, J.I., Muguerza, J., Perez, J.M., and Perona I., (2009), SIHC: A Stable Incremental Hierarchical Clustering Algorithm, ICEIS, pp.300-304.
[6] Hossein Zadeh, P.D., and Reformat, M.Z., (2013), Semantic Similarity Assessment of Concepts Defined in Ontology, Inf. Sci., 250, pp.21-39.
[7] Hossein Zadeh, P.D., and Reformat, M.Z., (2012), Context-aware Similarity Assessment within Semantic Space Formed in Linked Data, J. Ambient Intell. Humaniz. Comput., 4, pp.515-532.
[8] Lalithsena, S., Hitzler, P., Sheth, A., and Jain, P., (2013), Automatic Domain Identification for Linked Open Data, IEEE/WIC/ACM Inter. Conf. on Web Intelligence and Intelligent Agent Technology, pp.205-212.
[9] Szekely, G.J., and Rizzo, M.L., (2005), Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method, J. Classif., 22, pp.151-183.
[10] Zadeh, L.A, (1965), Fuzzy sets, Information and Control, 8, pp.338-353.
[11] Zadeh, L.A, (1984), Coping with the imprecision of the real world, Commun. of the ACM, 27, pp.304-311.
[12] Zong, N., Im, D.H., Yang, S., Namgoon, H., and Kim, H.G., (2012), Dynamic Generation of Concepts Hierarchies for Knowledge Discovering in Bio-medical Linked Data Sets, ICUIMC 12: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, Article 12, pp.1-5.
[13] http://www.w3.org/RDF/ (accessed Jan 30th, 2021).

Marek Z. Reformat - received his M.Sc. degree (with honors) from Technical University of Poznan, Poland, and Ph.D. from University of Manitoba, Canada. He is a Full Professor and Associate Chair of the Department of Electrical and Computer Engineering, University of Alberta. His research activities focus on development of methods and techniques for intelligent data modeling and analysis leading to translation of data into knowledge. He uses the concepts of Computational Intelligence - with fuzzy computing and possibility theory in particular -

- as key elements necessary for capturing relationships between pieces of data and knowledge, and for introducing human aspects to data analysis and decision-making processes resulting in more human-aware and human-like systems. He has published over 100 peer-reviewed publications in the areas of computational intelligence, knowledge and software engineering. He is an Associate Editor of a number of international journals. He is a past president of the North American Fuzzy Information Processing Society (NAFIPS), and a president of the International Fuzzy Systems Association (IFSA).

Ronald R. Yager - has worked in the area of machine intelligence for over twenty-five years. He has published over 500 papers and more then thirty books in areas related to fuzzy sets, decision-making under uncertainty and the fusion of information. He is among the world's top 1% most highly cited researchers with over 80,000 citations. He was the recipient of the IEEE Computational Intelligence Society's highly prestigious Frank Rosenblatt Award in 2016. He was the recipient of the IEEE Systems, Man and Cybernetics Society 2018 Lotfi Zadeh Pioneer Award. He was also the recipient of the IEEE Computational Intelligence Society Pioneer award in Fuzzy Systems.

He received honorary doctorates from the Azerbaijan Technical University, the State University of Information Technologies, Sofia Bulgaria and the Rostov on the Don University, Russia. Dr. Yager is a fellow of the IEEE, the New York Academy of Sciences and the Fuzzy Systems Association. He was given a lifetime achievement award by the Polish Academy of Sciences for his contributions. He is a Distinguished Adjunct Professor at King Abdulaziz University, Jeddah, Saudi Arabia. He was a professor at Iona College and is director of the Machine Intelligence and is now a candidate for Emeritus Professor. He served at the National Science Foundation as program director in the Information Sciences program. He was a NASA/Stanford visiting fellow and a research associate at the University of California, Berkeley. He has been a lecturer at NATO Advanced Study Institutes. He was a distinguished honorary professor at the Aalborg University Denmark. He was distinguished visiting scientist at King Saud University, Riyadh, Saudi Arabia. He received his undergraduate degree from the City College of New York and his Ph. D. from the Polytechnic University of New York. Currently, he is Director of the Machine Intelligence Institute and Professor of Information Systems at Iona College. He was editor and chief of the International Journal of Intelligent Systems. He serves on the editorial board of numerous technology journals.

Jesse Xi Chen - is a Ph.D student at the University of Alberta. He obtained his Master degree in Software and Intelligent Systems also from the University of Alberta. His research interests include technologies of Semantic Web and Knowledge Graphs. He has designed and developed a software application targeting the discovery of concepts and their similarity in data represented as graphs in RDF (Resource Description Framework) format. He spent five months as a research intern in the National Institute of Informatics, Tokyo, Japan, where he worked on formalizing a process of converting statistical data into RDF with emphasis on conforming to existing standards and re-usability.