Acta Cybernetica **26** (2023) 243–266.

Refined Fuzzy Profile Matching[*]

Gábor Rácz[ab], Attila Sali[ac], and Klaus-Dieter Schewe[d]

Abstract

A profile describes a set of properties, e.g. a set of skills a person may have or a set of skills required for a particular job. Profile matching aims to determine how well a given profile fits to a requested profile and vice versa. Fuzzyness is naturally attached to this problem. The filter-based matching theory uses filters in lattices to represent profiles, and matching values in the interval [0,1], so the lattice order refers to subsumption between the concepts in a profile. In this article the lattice is extended by additional information in form of weighted extra edges that represent partial quantifiable relationships between these concepts. This gives rise to fuzzy filters, which permit a refinement of profile matching. Another way to introduce fuzzyness is to treat profiles as fuzzy sets. In the present paper we combine these two aproaches. Extra edges may introduce directed cycles in the directed graph of the ontology, and the structure of a lattice is lost. We provide a construction grounded in formal concept analysis to extend the original lattice and remove the cycles such that matching values determined over the extended lattice are exactly those resulting from the use of fuzzy filters in case of crisp profiles. For fuzzy profiles we show how to modify the weighting construction while eliminating the directed cycles but still regaining the matching values. We also give sharp estimates for the growth of the number of vertices in this construction.

Keywords: lattice, filter, matching measure, fuzzy sets, fuzzy filter, lattice enrichment, formal concept analysis

[*]Research of the second author was partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–116769 and SNN-135643. This work was also supported by the BME-Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC) and by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Artificial Intelligence National Laboratory of Hungary.

[a]Alfréd Rényi Institute of Mathematics, Budapest, Hungary

[b]E-mail: gabee33@gmail.com, ORCID: 0000-0002-0733-1350

[c]E-mail: sali.attila@renyi.hu, ORCID: 0000-0002-4837-6360

[d]Zhejiang University, Hangzhou, Zhejiang, China, E-mail: kdschewe@acm.org, ORCID: 0000-0002-8309-1803

DOI: 10.14232/actacyb.277380

1 Introduction

A profile describes a set of properties, and profile matching is concerned with the problem of determining how well a given profile fits to a requested one. Profile matching appears in many application areas such as matching applicants for jobs to job requirements, matching system configurations to requirements specifications, matching team players to game strategies in sport, etc.

A simple approach to profile matching considers profiles as sets of unrelated items, which leads to measuring the similarity or distance of sets. Several ways of definition of distances of sets were introduced such as Jaccard or Sørensen-Dice measures [17], which turned out to be useful in ecological applications. However, skills or properties included in profiles are usually not totally unrelated items, dependencies between them exist and need to be taken into account. For example, in the human resources area several taxonomies for skills, competences and education such as DISCO [6], ISCED [13] and ISCO [14] have been set up. These taxonomies organize the individual properties into a lattice structure. Popov and Jebelean [26] proposed to define an asymmetric matching measure on the basis of filters in such lattices. They represented a profile P with the lattice filter generated by P on the basis that having a specialized skill imples the having a more general skill like knowledge of Java assumes knowledge of Object Oriented Programming as in Figure 4.

Besides such subsumption relationships captured by the lattice order other "horizontal" relationships exist as well. For instance, a job applicant may have some other skills with certain probabilities or of some (not complete) proficiency level, e.g. we may reasonably assume that knowledge of Java implies knowledge of NetBeans up to a grade of 0.7 (or with probability 0.7). This kind of dependencies are exploited in [27]. The idea is that a given profile is considered better than another one for a given requested profile, if they match equally using the filter-based measure, but the first one has more items implied partially that match the requested profile. In this way we get a refinement of the filter-based matchings using the maximum weight of a path from the profile's nodes to a vertex x. This process results in a set of nodes with grades in [0,1], which can be interpreted as a fuzzy set. Actually, it turns out to be a fuzzy filter [12, 18].

However, the introduction of extra edges may give rise to directed cycles, and the elegance of the uniform filter-based matchings is destroyed. Therefore, we raised the question in [28], if the extra edges can be used to modify the original lattice in such a way that instead of using fuzzy filters ordinary filters in the modified lattice can be exploited, which means that the refinement can be re-interpreted in the context of the filter-based matching theory. The answer to this problem is positive, as we explore in this article.

1.1 Our Contribution

In this article we develop an enriched theory of profile matching centered around the idea from [27] using weighted extra edges in addition to edges defined by the order in a lattice to capture partial relationships between concepts in a profile. How matching measures can be extended has been shown in our previous work [27].

We now provide a construction that gets rid of directed cycles caused by the extra edges. In doing so we show that all matching results that can be obtained by exploiting extra edges can also be obtained from an extended lattice without such extra edges. That is, the theory of profile matching remains within the filter-based approach that we developed in [21], which underlines the power and universality of this theory. In particular, we emphasize how to obtain the lattices underlying the matching theory from knowledge bases that define concepts used in given and requested profiles, and accordingly we call the lattices also *ontology lattices*. These knowledge bases are grounded in description logics, so the lattice extensions provide also feedback for fine-tuning the knowledge representation, whereas weighted extra-edges are not supported in the knowledge bases. In [21] it was also shown that under mild plausibility constraints on human-defined matchings appropriate weights can be defined such that the filter-based matchings preserve the human-defined rankings, which further enables linear optimization to synchronize matchings with human expertise. These results on learning matchings from human expertise can now be carried over to the refined matching theory.

The extension is done by extending the original ontology lattice by new nodes and weighting of the nodes. The result is a directed acyclic graph, whose structure reflects the different possible path lengths between nodes of the ontology lattice. A directed acyclic graph naturally represents a poset, although not a lattice in general. In order to gain back the lattice structure formal concept analysis is used.

The concept of offers and applications from [27, 28] is extended to fuzzy sets. That is we interpret such formulations as "knowledge of skill X is an advantage" by giving a membership value to skill X in the offer a number from $(0, 1)$, measuring the importance of X. Similarly, applications are also considered as fuzzy sets where the membership values signify the proficiency of the applicant in the given skill.

While the extension of given profiles is natural, e.g. for job applications the consideration of skills derived from extra edges appears natural, as employers may benefit from these skills, it is not so clear whether the requested profiles should be extended as well. On one hand, profiles should be handled uniformly, as they could represent both given and requested profiles. On the other hand, if requested profiles, e.g. requirements in job offers, are also extended, then it may happen that a high matching score may result only from derived skills, not from the ones originally required, which may be considered as being misleading and disadvantageous. In the present paper we discuss both scenarios, the latter one being treated by applying different weighting functions for given and requested profiles.

Note the conceptual difference between horizontal connections represented by extra edges and the membership values of skills in offers and applications. The extra edges belong to the taxonomy used and are determined by the domain experts, while fuzzy values are determined by the firms and individuals who apply the matching measure to rank applications for offers.

1.2 Organization of the Article

The remainder of this article is organized as follows. In Section 2 we provide a brief introduction of the fundamentals of filter-based profile matching as developed in our previous research (summarized in [21]), and then extend the approach by using extra edges and fuzzy filters. Section 3 is then dedicated to the construction of the lattice enlargement using formal concept analysis and the proof that matching values using extra edges can be equivalently obtained by ordinary matching values on the extended lattice. We also give node weightings that preserve the weights of fuzzy filters assuming that requested profiles are also extended. Section 4 contains the analysis for the case that requested profiles are not extended. Section 3.3 discusses related extremal problems concerning how the size of the constructed enlargement relates to the size of the original lattice. It is included for the sake of completeness, the proofs of the statements can be found in [28]. Finally, in Section 5 we discuss related work, and in Section 6 we conclude the article with a brief summary.

2 Profile Matching Based on Lattices and Filters

In this section we briefly present the definitions underlying the matching theory from [21] and its refinement from [27], as well as notations of fuzzy set theory used. Matching theory is based on lattice $\mathcal{L}$, and a profile is represented by a filter $\mathcal{F}$ in the lattice. A *matching measure* is a function defined on pairs of filters. If μ is such a matching measure and $\mathcal{F}$, $\mathcal{G}$ are filters, then $\mu(\mathcal{F}, \mathcal{G})$ will be a real number in the interval $[0, 1]$, which is called a *matching value*. Matching measures in general exploit weights assigned to concepts in the lattice $\mathcal{L}$.

Let $\mathcal{L}(S, \leq)$ be a lattice. Informally, for $A, B \in S$ we have $A \leq B$, if the property A subsumes property B, e.g. for skills this means that a person with skill A will also have skill B. A *filter* is a non-empty subset $\mathcal{F} \subseteq S$, such that for all C, C' with $C \leq C'$ whenever $C \in \mathcal{F}$ holds, then also $C' \in \mathcal{F}$ holds.

Let $\mathbb{F} \subseteq \mathcal{P}(S)$ denote the set of filters. A *weighting function* on S is a function $w: \mathcal{P}(S) \to [0, 1]$ satisfying (1) $w(S) = 1$, and (2) $w(\bigcup_{i \in I} A_i) = \sum_{i \in I} w(A_i)$ for pairwise disjoint A_i $(i \in I)$.

Definition 1. A *matching measure* is a function $\mu: \mathbb{F} \times \mathbb{F} \to [0, 1]$ such that $\mu(\mathcal{F}_1, \mathcal{F}_2) = w(\mathcal{F}_1 \cap \mathcal{F}_2)/w(\mathcal{F}_2)$ holds for some weighting function w on $\mathcal{L}$.

The matching measure μ_{pj} defined in [26] uses simply cardinalities:

$$\mu_{pj}(\mathcal{F}_1, \mathcal{F}_2) = \#(\mathcal{F}_1 \cap \mathcal{F}_2)/\#\mathcal{F}_2$$

Thus, it is defined by the weighting function w on S with $w(A) = \#A/\#\mathcal{L}$, i.e. all properties have equal weights. From Section 3 onwards we will tacitly assume that properties have equal weight. This will simplify our presentation, and the extension of our theory to matching measures with general weighting functions is straightforward.

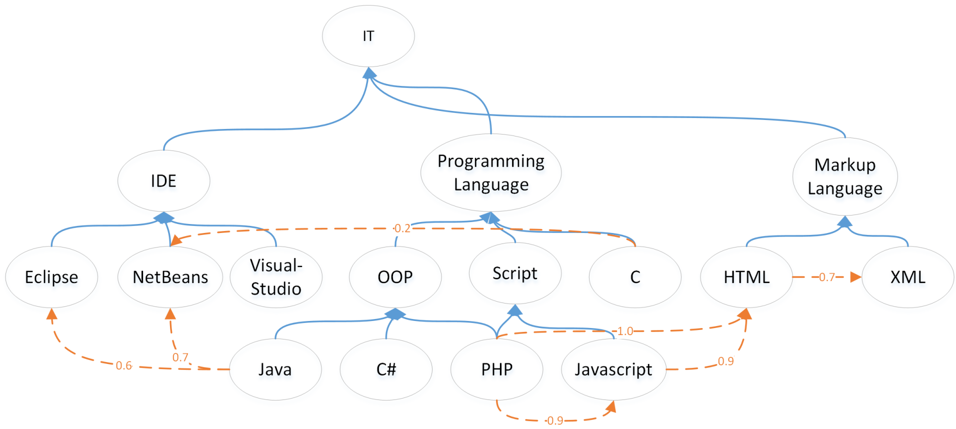

Figure 1: Fragment of a graph with lattice edges (solid) and extra edges (dashed) and assignment of degrees.

Let $\mathbf{F}(S)$ be the collection of fuzzy sets over S. For an $X \in \mathbf{F}(S)$ and $s \in S$ let $\mu_X(s)$ denote the membership value of s in X. We also write $X = \{x_1 : \gamma_1, x_2 : \gamma_2, \ldots, x_n : \gamma_n\}$ where $\gamma_i = \mu_X(x_i)$. The *support* of fuzzy set $X \in \mathbf{F}(S)$ is $\mathsf{supp}(X) = \{s \in S | \mu_X(s) > 0\}$. For two fuzzy sets F, G of $\mathbf{F}(S)$ let $F \cap G = \{s : \gamma_s | s \in S$ and $\gamma_s = \min\{\mu_F(s), \mu_G(s)\}$, furthermore let $||F|| := \sum_{v:\gamma_v \in F} \gamma_v$, i.e. $||\cdot||$ denotes sigma cardinality and intersection is defined as the *min* t-norm. Note, that other cardinality and intersection functions could be applied in the same way [35, 12]. We assume that $v : \gamma_v \in F$ means $\gamma_v = \mu_F(v) > 0$.

We can extend the lattice with additional information in form of so called extra edges that represent some kind of quantifiable relationship between skills. However, these edges can form cycles in the hierarchy therefore we use directed graphs to handle them instead of the lattice structure [27].

Let $G = (V, E)$ be a directed graph where $V = S$ and $E = E_{lat} \cup E_{ext}$ is a set of lattice edges and extra edges such that for two nodes $v_i, v_j \in V : (v_i, v_j) \in E_{lat}$ iff v_j covers v_i, i.e. $v_i < v_j$ and there exists no v_k such that $v_i < v_k < v_j$. Furthermore, $(v_i, v_j) \in E_{ext}$ iff there is an extra edge between v_i and v_j. Let $w_{edge} : E \to [0, 1]$ be an edge weighting function such that for all $e_{lat} \in E_{lat} : w_{edge}(e_{lat}) = 1$ and for all $e_{ext} \in E_{ext} : w_{edge}(e_{ext}) \in [0, 1]$ that represents the strength of the relationship between start and end node of the edge. See Figure 1 for a fragment of such a graph. Let $p_F(x, v)$ denote the set of directed paths from node x to node v using edges of a subset $F \subseteq E$ of edge set E of G.

Let application A and offer O be fuzzy sets over S and define a matching function of an application A to an offer O using the graph in the following way. First, we define function *ext* to extend the application and the offer with all the skills that are available from them via directed path in G. For an arbitrary fuzzy set of skills

$X \in \mathbf{F}(S)$ and a subset $F \subseteq E$ of edges let

$$ext_F(X) = \{v\colon \gamma_v | v \in S \text{ and } \gamma_v = max_{x', p \in p_F(x', v)} length(p) \cdot \mu_X(x')\}, \qquad (1)$$

where length of a path $p = (v_1, \ldots, v_n)$ is the product of the edge weights on p, i.e. $length(p) = \prod_{i=1}^{n-1} w_{edge}((v_i, v_{i+1}))$ and if $p_F(x', v) = \emptyset$, then naturally $length(p) = 0$ for $p \in p_F(x', v)$.

Fuzzy filters were introduced in [18]. A fuzzy set Y over S is a fuzzy filter in $\mathcal{L} = (S, \leq)$ if for all $t \in [0, 1]$ the level set $Y_t = \{y \in Y : \mu_Y(y) \geq t\}$ is a filter in $\mathcal{L}$. A crisp version of the following was proven in [28].

Theorem 1. *Let $G = (S, E = E_{lat} \cup E_{ext})$ be a directed graph with edge weights $w_{edge}\colon E \to [0, 1]$ extending the lattice $\mathcal{L}(S, \leq)$, and let $X \in \mathbf{F}(S)$ be a fuzzy set over S. Then the extension $ext_E(X)$ of X with respect to E is a fuzzy filter in $\mathcal{L}$.*

Proof. Let $s \in ext_E(X)_t$ and $s < s'$ in $\mathcal{L}$. Furthermore, let $x \in S$ and $p \in p_E(x, s)$ where the maximum in (1) is taken. Since $s < s'$ in $\mathcal{L}$, there exists a directed path p' from s to s' using only lattice edges. The concatenation of p and p' is a directed *walk* q in G from x to s' such that $length(p) = length(q)$, because lattice edges have weight 1. Let q' be the the walk from x to s' of fewest edges such that $q' \subseteq q$. Then clearly $q' \in p_E(x, s')$ and $length(q') \geq length(q) = length(p)$. Hence, $\gamma_{s'} \geq \gamma_s$ implying that $s' \in ext_E(X)_t$. □

Example 1. For the graph in Figure 1 take the following fuzzy sets of skills

$$O = \{Java\colon 1.0, Netbeans\colon 0.9, XML\colon 0.5\} \text{ and}$$
$$A = \{Java\colon 1.0, PHP\colon 0.9, Eclipse\colon 0.7\}$$

These generate the following fuzzy filters:

$$\begin{aligned} ext_E(O) = \{&Java\colon 1.0, Netbeans\colon 0.9, XML\colon 0.5, OOP\colon 1.0, PL\colon 1.0, \\ &IT\colon 1.0, IDE\colon 0.9, Eclipse\colon 0.8, ML\colon 0.5\} \\ \text{and} \qquad ext_E(A) = \{&Java\colon 1.0, PHP\colon 0.9, Eclipse\colon 0.8, OOP\colon 1.0, PL\colon 1.0, \\ &IT\colon 1.0, Script\colon 0.9, IDE\colon 0.8, Netbeans\colon 0.7, \\ &Javascript\colon 0.81, HTML\colon 0.9, ML\colon 0.9, XML\colon 0.63\} \end{aligned}$$

This gives rise to the intersection fuzzy filter

$$\begin{aligned} ext_E(A) \cap ext_E(O) = \{&Java\colon 1.0, OOP\colon 1.0, PL\colon 1.0, IT\colon 1.0, IDE\colon 0.8, \\ &Eclipse\colon 0.8, Netbeans\colon 0.7, XML\colon 0.5, ML\colon 0.5\} \end{aligned}$$

Assuming a weighting function w that assigns the same weight to all elements we obtain the matching value $\mu(ext_E(A), ext_E(O)) = \frac{6}{7.1}$.

It perfectly makes sense to use lattice edges to extend applications and offers as lattice edges describe specialization relation between skills. Namely if an applicant possesses a special skill then he or she must possess the more general skills as well. However extra edges are used in the extension as well to get more selective matching functions that help differentiate applications.

Let us call nodes in $\mathsf{supp}(ext_E(X)) \setminus \mathsf{supp}(ext_{E_{lat}}(X))$ *derived* nodes for a fuzzy set $X \in \mathbf{F}(S)$ of skills. We investigate two approaches or philosophies when extending profiles using the extra edges. The first one is *symmetric*, that is the case when offers and applications are treated in the same way. In this case we use extension function ext_E for both, offers O and applications A. The advantage is that we only have to apply one weighting function and the proof of equivalence of different representations is simpler than that of the other case. There is a disadvantage, though. If offers are also extended with derived skills, then an application may obtain high matching value just having those skills. However, it is not really advantageous for an employer, as required skills are not in the application.

The second approach called the *strict approach* is when offers are only extended with non-derived nodes, that is ext_E is used for applications but $ext_{E_{lat}}$ is used for offers. This is the approach of [27]. The disadvantage of this case is that different weighting functions have to be applied for applications and offers, consequently the proofs of equivalences are more complicated. However, the point of view of employers is better represented in the second way. An application has to have good matching in target skills to score high, and the derived skills can be used to rank applications scoring equally otherwise. Note, that $\mathsf{supp}(ext_{E_{lat}}(X))$ is exactly the set of nodes contained in the lattice filter generated by the support $\mathsf{supp}(X) = \{s \in S \colon \mu_X(s) > 0\}$ in the ontology lattice $(S, \leq)$.

We adapted the profile matching function proposed by Popov et. al. [26] to fuzzy sets in [27]. We use the same function here except the different approaches in extension of offers. So, let the matching value of A to O be

$$match_{sym}(A, O) = \frac{||ext_E(A) \cap ext_E(O)||}{||ext_E(O)||} \tag{2}$$

in case of the symmetric approach, and

$$match(A, O) = \frac{||ext_E(A) \cap ext_{E_{lat}}(O)||}{||ext_{E_{lat}}(O)||} \tag{3}$$

in case of the strict approach.

3 Lattice Enlargement

In this section, we present a graph transformation method to eliminate extra edges from extended lattices preserving symmetric matching values of applications to offers, and then we use formal concept analysis to restore lattice properties in the transformed graphs.

3.1 Extension Graph

Let $G = (V, E)$ be a directed graph with weighting function w_{edge} as defined above and c_{ij} be the length of the longest path from v_i to v_j where $v_i, v_j \in V$ are two nodes. Let $v_{i_1 j}, \ldots v_{i_k j}$ be the nodes from where v_j is available via directed path such that $c_{i_1 j} \leq \cdots \leq c_{i_k j}$. Let $c^{j_1}, \ldots, c^{j_{l_j}}$ denote the different values among $c_{i_1 j}, \ldots, c_{i_k j}$, i.e. $c^{j_1} < \cdots < c^{j_{l_j}}$.

For all $c^{j_1} \ldots c^{j_{l_j}}$, add new nodes $V_j = \{v_{j_1}, \ldots, v_{j_{l_j - 1}}\}$ (for simplicity let $v_{j_{l_j}} = v_j$) to V and add new edges of weight one from $v_{j_{l_j}}$ to $v_{j_{l_j - 1}}$, ..., from v_{j_2} to v_{j_1}, and from v_{j_1} to the top to E. The new edges form a directed path from v_j to the top. Let $q_j = (v_{j_{l_j}}, \ldots, v_{j_1}, top)$ denote that path. Assign weight $w_{j_k} = c^{j_k} - c^{j_{k-1}}$ to v_{j_k} $(k = 1, \ldots, l_j)$ where $c^{j_0} = 0$. Note, that $\sum_{k=1}^{l} w_{j_k} = 1$ as it is a telescopic sum. If the length of the longest path from v_i to v_j was c^{j_k}, then add a new edge of weight one from v_i to v_{j_k}. Finally, remove all extra edges from the graph. Now each edge has weight one, so edge weights can be ignored.

Let $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ denote the modified graph, called *extension graph*, and w_{node} denote the node weighting function defined.

New nodes of V_j and new edges of q_j can be considered as an extension of v_j to a chain because there do not start edges from intermediate nodes to other chains so out-degrees of intermediate nodes are always one. We call v_j the *base node* of the chain. Base nodes of such chains are nodes of $\mathcal{L}$, and G as well.

Let q_j and q_k be two chains with base nodes v_j and v_k, respectively. Then, an edge from q_k to q_j in G' can go

- from v_k to v_j and then it represents a directed path in G from v_k to v_j containing lattice edges only;

- from v_k to an intermediate node v_{j_i} of q_j and then it represents a directed path $p_{v_k v_j}$ of G from v_k to v_j such that $length(p_{v_k v_j}) = \sum_{s=1}^{i} w_{node}(v_{j_s})$.

Note, that lattice edges in G are acyclic so the corresponding edges in G' are acyclic as well, and newly added edges between different chains start from base nodes of chains only. So G' is an acyclic graph.

Figure 2 shows an example of the construction of G'. There is the original graph, called G, on the left. Blue (solid) edges represent lattice edges and orange (dashed) edges with numbers on them represent extra edges and their weights. There is the extension graph, called G', on the right where green edges represent the newly added edges, and numbers in the top right corners of nodes are weights of the nodes.

As it can be seen, for example, node A of G has been transformed into the chain $q_A = (A, A_1, Top)$ since A is available via lattice edges (i.e. via maximum length paths) from $B, C, Bottom$ and it is available from D via the path $p_{DA} = (D, C, A)$ whose length is 0.8 and A is not available from any other nodes. Therefore A_1 got the weight 0.8 and A got the weight 0.2.

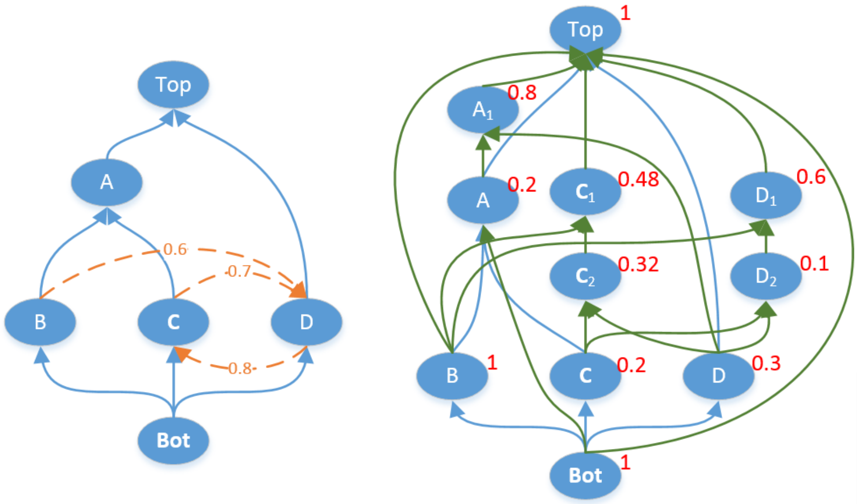

Figure 2: Lattice with extra edges and the generated extension graph

The extension graph defined above makes calculating matching value of *crisp* offers and applications easy. The following was proven in [28], we include the proof here for further use and sake of completeness.

Lemma 1. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph. Let $O \subseteq S$ be an offer and $A \subseteq S$ be an application, that are crisp sets. Then,*

$$match_{sym}(A, O) = \frac{||ext_E(A) \cap ext_E(O)||}{||ext_E(O)||} = \frac{||ext_{E'}(A) \cap ext_{E'}(O)||}{||ext_{E'}(O)||}, \qquad (4)$$

where $ext_{E'}(X)$ denotes the set of vertices of G' that are reachable from some nodes in X via directed paths of G'.

Proof. Let $u \in G'$ and let $q_z = (z_{l_z}, \ldots, z_1, top)$ be the node chain with base node $z \in G$ that contains u, i.e. $z_{l_z} = z$ and $u = z_i$ for some $i \in [1 \ldots l_z]$. First, we will show for an arbitrary $X \subseteq S$ that $u \in ext_{E'}(X)$ iff $z \in ext_E(X)$.

If $u \in ext_{E'}(X)$, then there is a node $a \in X \subseteq V'$ and a directed path $p_{au} = (x_1, \ldots, x_i, x_{i+1}, \ldots x_n)$ from a to u in G' where $x_1 = a$ and $x_n = u$. If $a = z$ then $z \in ext_E(X)$. Otherwise let $x_{i+1} = z_m$ be the first node of p_{au} that is an intermediate node of q_z as well. Then for $j \in [1 \ldots i-1]$: x_j, x_{j+1} are nodes of G, and edges (x_j, x_{j+1}) of p_{au} represent directed paths containing lattice edges only in G. Note that lattice edges form an acyclic subgraph of G. Therefore the concatenations of lattice edge paths $p_{x_j x_{j+1}}$ represented by directed edges (x_j, x_{j+1}) of G' for $j \in [1 \ldots i-1]$ is a path p_{ax_i} in the lattice $\mathcal{L}$ from a to x_i. Now, the edge $(x_i, x_{i+1} = z_m)$ of G' represents a directed path $p_{x_i z}$ from x_i to z in G using some extra edges. The concatenation of p_{ax_i} and p_{x_iz} is a directed walk from a to z in G, so it contains a directed path p_{az}, that is $z \in ext_E(X)$.

On the other hand, if $z \in ext_E(X)$ with grade γ_z, then there is a node $b \in X$ and a maximal length path p_{bz} from b to z in G such that $length(p_{bz}) = \gamma_z$. In that case, there is an edge from b to z_r in G' for some $r \in [1..l_z]$ such that $\sum_{s=1}^{r} w'_{node}(z_r) = length(p_{bv})$ and also $z_r, z_{r-1}, z_1 \in ext_{E'}(X)$.

Consequently, $ext_{E'}(A) \cap ext_{E'}(O)$ contains fragments of chains generated from base nodes that are available from both A and O in G. Sum of node weights in a fragment equals to the minimum of the lengths of the maximal length paths starting from A or O ending in the base node of the chain. Thus, $||ext_E(A) \cap ext_E(O)|| = ||ext_{E'}(A) \cap ext_{E'}(O)||$ and $||ext_E(O)|| = ||ext_{E'}(O)||$, i.e. equation (4) holds. $\square$

If offers O and applications A are allowed to be fuzzy sets, that is $O, A \in \mathbf{F}(S)$, then the situation is more complicated. As an example consider the lattice and extension graph of Figure 2. If $A = \{D{:}\,0.6\}$ and $O = \{B{:}\,0.9\}$, then $ext_E(A) = \{A{:}\,0.48, C{:}\,0.48, D{:}\,0.6, top{:}\,0.6\}$ and $ext_E(O) = \{A{:}\,0.9, B{:}\,0.9, D{:}\,0.54, C{:}\,0.432, top{:}\,0.9\}$, so $ext_E(A) \cap ext_E(O) = \{A{:}\,0.48, C{:}\,0.432, D{:}\,0.54, top{:}\,0.6\}$. Observe that for any $X \in \mathbf{F}(S)$ we have $\mathsf{supp}(ext_E(X)) = ext_E(\mathsf{supp}(X))$ that would suggest defining $ext'_{E'}(X) = \{v{:}\,\gamma_v | \gamma_v = \max \mu_X(x) w_{node}(v)$ for $x \in \mathsf{supp}(X)$ and $\exists$ directed path from x to v in $G'\}$. However, this definition would give

$$ext'_{E'}(A) = \{A_1{:}\,0.48, C_2{:}\,0.192, C_1{:}\,0.288, D{:}\,0.18, D_2{:}\,0.06, D_1{:}\,0.36, top{:}\,0.6\}$$

and

$$ext'_{E'}(O) = \{A{:}\,0.18, A_1{:}\,0.72, B{:}\,0.9, C_1{:}\,0.432, D_1{:}\,0.54, top{:}\,0.9\}$$

resulting in

$$ext'_{E'}(A) \cap ext'_{E'}(O) = \{A_1{:}\,0.48, C_1{:}\,0.288, D_1{:}\,0.36, top{:}\,0.6\}.$$

Thus, $||ext'_{E'}(A) \cap ext'_{E'}(O)|| \neq ||ext_E(A) \cap ext_E(O)||$.

In order to resolve this problem we charge the contributions of nodes of each chain to the chain's top node as follows. For $x, v \in S$ define $t(x, v) = \sum_{i=1}^{m} w_{node}(v_i)$ where (x, v_m) is the edge of the extension graph G' from x to the chain q_v. If no such edge exists then $t(x, v)$ is defined to be 0. Note, that values $t(x, v)$ can be calculated as a preprocessing step for every pair $x, v \in S$, since they do not depend on particular profiles. Let $X \in \mathbf{F}(S)$ and $x \in \mathsf{supp}(X)$, furthermore let $ext^f_{E'}(X) = \{v_1{:}\,\gamma_{v_1} | v \in S$ and $\gamma_{v_1} = \max_{x \in \mathsf{supp}(X)} \mu_X(x) t(x, v)\}$. Considering the previous example of $A = \{D{:}\,0.6\}$ and $O = \{B{:}\,0.9\}$, we obtain $ext^f_{E'}(A) = \{A_1{:}\,0.48, C_1{:}\,0.48, D_1{:}\,0.6, top{:}\,0.6\}$ and $ext^f_{E'}(O) = \{A_1{:}\,0.9, B_1{:}\,0.9, D_1{:}\,0.54, C_1{:}\,0.432, top{:}\,0.9\}$. Note that $B_1 = B$ and $top_1 = top$ as their chains contain one element, respectively.

Theorem 2. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph. Let $O \in \mathbf{F}(S)$ be an offer and $A \in \mathbf{F}(S)$ be an application . Then*

$$match_{sym}(A, O) = \frac{||ext_E(A) \cap ext_E(O)||}{||ext_E(O)||} = \frac{||ext^f_{E'}(A) \cap ext^f_{E'}(O)||}{||ext^f_{E'}(O)||}. \qquad (5)$$

Proof. There is a directed edge (x, v_m) from a node $x \in S$ to $v_m \in V'$ iff there exists a directed path from x to v in G by the construction of the extension graph. Furthermore, $\sum_{i=1}^{m} w_{node}(v_i)$ the length of the longest path from x to v in G. Thus, for any $X \in \mathbf{F}(S)$ we have $v{:}\, \gamma_v \in ext_E(X) \iff v_1{:}\, \gamma_v \in ext^f_{E'}(X)$.This together with $v \in \mathsf{supp}(ext_E(A) \cap ext_E(O)) \iff v \in \mathsf{supp}(ext_E(A)) \cap \mathsf{supp}(ext_E(O)) \iff v \in \mathsf{supp}(ext^f_{E'}(A)) \cap \mathsf{supp}(ext^f_{E'}(O)) \iff v \in \mathsf{supp}(ext^f_{E'}(A) \cap ext^f_{E'}(O))$ completes the proof. $\square$

Note, that G' is acyclic by its construction but does not necessarily define a lattice. There is a natural way to define a lattice, namely a concept lattice from G' in which matching values of *crisp* applications to *crisp* offers are preserved.

3.2 Concept Lattices

First, we define a formal context and formal concepts based on G'. Let (V'_1, V'_2, T') be a formal *context*, where $V'_1 = V'_2 = V'$ and $(v_i, v_j) \in T'$ iff v_j is available from v_i via directed path supposing that the relation is reflexive. Consider the element of V'_1 as start points and the element of V'_2 as end points of directed paths in G'. Let $I \subseteq V'_1$ and $J \subseteq V'_2$ and let us define their dual sets I^{D_s} and J^{D_e} as follows:

$$
\begin{aligned}
I^{D_s} &= \{b \in V'_2 \mid (a, b) \in T' \text{ for all } a \in I\} \\
J^{D_e} &= \{a \in V'_1 \mid (a, b) \in T' \text{ for all } b \in J\}
\end{aligned}
$$

A *concept* of the context (V'_1, V'_2, T') is a pair $\langle I, J \rangle$ such that $I \subseteq V'_1$, $J \subseteq V'_2$ and $I^{D_s} = J$, $J^{D_e} = I$. I is called an *extent* of $\langle I, J \rangle$, and J is called an *intent* of $\langle I, J \rangle$.

Table 1: Formal context (V'_1, V'_2, T')

	Bot	B	C	C1	C2	D	D1	D2	A	A1	Top
Bot	X	X	X	X	X	X	X	X	X	X	X
B		X		X			X		X	X	X
C			X	X	X		X	X	X	X	X
C1				X							X
C2				X	X						X
D				X	X	X	X	X		X	X
D1							X				X
D2							X	X			X
A									X	X	X
A1										X	X
Top											X

Table 1 shows the formal context (V'_1, V'_2, T') that was generated based on graph G' of Figure 2. Labels of rows and columns represent the elements of V'_1 and the elements of V_2', respectively. There is an X in row i column j if $(i,j) \in T'$, i.e. j is available from i via directed path in G'.

Lemma 2. *If G' is an acyclic graph, then*

1. *For every concept $\langle I, J\rangle$ of the context (V_1', V_2', T'): $I \cap J = \{v\}$ for some $v \in V'$ or $I \cap J = \emptyset$*
2. *For every $v \in V'$: there is a concept $\langle I_v, J_v\rangle$ in the context (V_1', V_2', T') such that $I_v \cap J_v = \{v\}$.*

Proof.

1. Indirectly, suppose that for a concept $\langle I, J\rangle$ of (V_1', V_2', T') and for two different nodes $u, v \in V'$: $u, v \in (I \cap J)$ holds. In this case $(u,v) \in T'$ and $(v,u) \in T'$ hold as well. It would mean that there is a cycle in G' which is a contradiction as G' is acyclic.
2. For a node $v \in V'$ let $J_v = \{v\}^{D_s}$ be the set of all nodes that are available from v via directed path (including v itself). Let $I_v = J_v^{D_e}$, then $v \in I_v$. If $I_v = \{v\}$, then $\langle I_v, J_v\rangle$ is the concept we are looking for.
Otherwise, suppose that for a node u such that $u \neq v$: $u \in I_v = J_v^{D_e} = (\{v\}^{D_s})^{D_e}$. That means $(u,v) \in T'$, i.e. v is available from u. As T' is a transitive relation $\{v\}^{D_s} \subseteq \{u,v\}^{D_s}$. However $\{u,v\}^{D_s} \subseteq \{v\}^{D_s}$ because $\{u,v\}^{D_s}$ cannot contain such node that is not available from all nodes of $\{u,v\}$. Following this construction we can get that if $J_v^{D_e} = I_v = \{u_1, \ldots, u_i, v\}$, then $I_v^{D_s} = \{u_1, \ldots, u_i, v\}^{D_s} = \{v\}^{D_s} = J_v$. Therefore $\langle\{u_1, \ldots, u_i, v\}, \{v\}^{D_s}\rangle$ is a concept such that $\{u_1, \ldots, u_i, v\} \cap \{v\}^{D_s} = \{v\}$.

□

Let $\mathcal{B}(V_1', V_2', T')$ be the set of all formal concepts in the context, and $\leq$ be a subconcept-superconcept order over the concepts such that for any $\langle A_1, B_1\rangle$, $\langle A_2, B_2\rangle \in \mathcal{B}(V_1', V_2', T')$: $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$, iff $A_1 \subseteq A_2$ (or, iff $B_2 \subseteq B_1$). $(\mathcal{B}(V_1', V_2', T'), \leq)$ is called *concept lattice* [10] and let $cl((\mathcal{L}, E_{ext}))$ denote the concept lattice obtained from the extension graph $ext(\mathcal{L}, E_{ext})$.

Figure 3 [1] shows concept lattice of the context (V_1', V_2', T') from Table 1. Concepts $\langle I_v, J_v\rangle$ where $I_v \cap J_v = \{v\}$ are labeled with v. For example, $\langle I_{C_2}, J_{C_2}\rangle = \langle\{Bot, C, C_2, D\}, \{C_2, C_1, Top\}\rangle$. But, concepts $\langle I, J\rangle$ such that $I \cap J = \emptyset$ are unlabeled like the $\langle\{Bot, B, C\}, \{A, A_1, C_1, D_1, Top\}\rangle$ parent of concepts B and C.

Another, larger example of concept lattice is shown on Figure 4 obtained from the ontology with added extra edges from [27] shown on Figure 1.

It is worth mentioning that the concept lattice $cl((\mathcal{L}, E_{ext}))$ generated from ontology $\mathcal{L}$ endowed with extra edges E_{ext} coincides with the Dedekind-McNeille completion [8] of the poset obtained as transitive closure of acyclic directed graph

[1] The concept lattices were generated using the Concept Explorer tool. Web page: http://conexp.sourceforge.net/

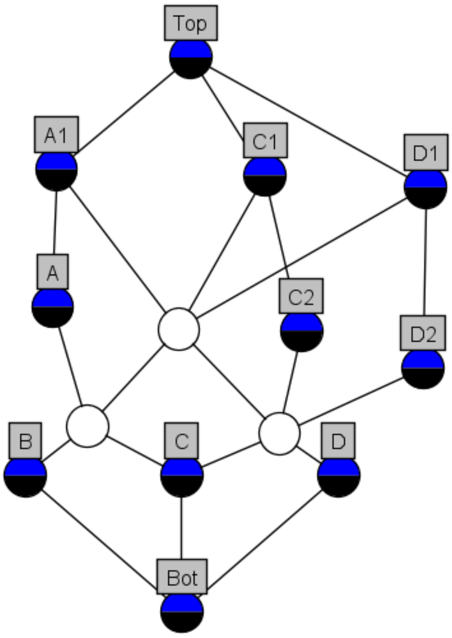

Figure 3: Concept lattice of context (V_1', V_2', T')

$ext(\mathcal{L}, E_{ext})$. Indeed, the collection of upper bounds of a subset S of elements of the poset is exactly the collection of the vertices reachable from the vertices of S via directed paths in the directed graph. We use the concept lattice formulation for two reasons. First, a direct construction is obtained skipping the step of constructing the poset from the directed graph $ext(\mathcal{L}, E_{ext})$. Second, the concept lattice structure allows us to define node weights properly.

A crisp offer $O = \{o_1, \dots, o_k\} \subseteq S = V \subseteq V'$ generates a filter $F_O \subseteq \mathcal{B}(V_1', V_2', T')$ in the concept lattice such that $F_O = \{\langle I, J\rangle \mid \exists \langle I_o, J_o\rangle \leq \langle I, J\rangle$ such that $I_o \cap J_o = \{o\}$ for some $o \in O\}$. Similarly, a crisp application A generates a filter F_A in the concept lattice.

Let $w_{con} \colon \mathcal{B}(V_1', V_2', T') \to [0, 1]$ be a concept weighting function such that for a concept $\langle I, J\rangle$ of $\mathcal{B}(V_1', V_2', T')$:

$$w_{con}(\langle I, J\rangle) = \begin{cases} w_{node} & \text{if } I \cap J = \{v\} \text{ for some } v \in V', \\ 0 & \text{otherwise.} \end{cases}$$

Let w_{fil} be a filter weighting function such that for a filter $F \in \mathcal{P}(\mathcal{B}(V_1', V_2', T'))$: $w_{fil}(F) = \sum_{\langle I,J\rangle \in F} w_{con}(\langle I, J\rangle)$.

The following was proven in [28].

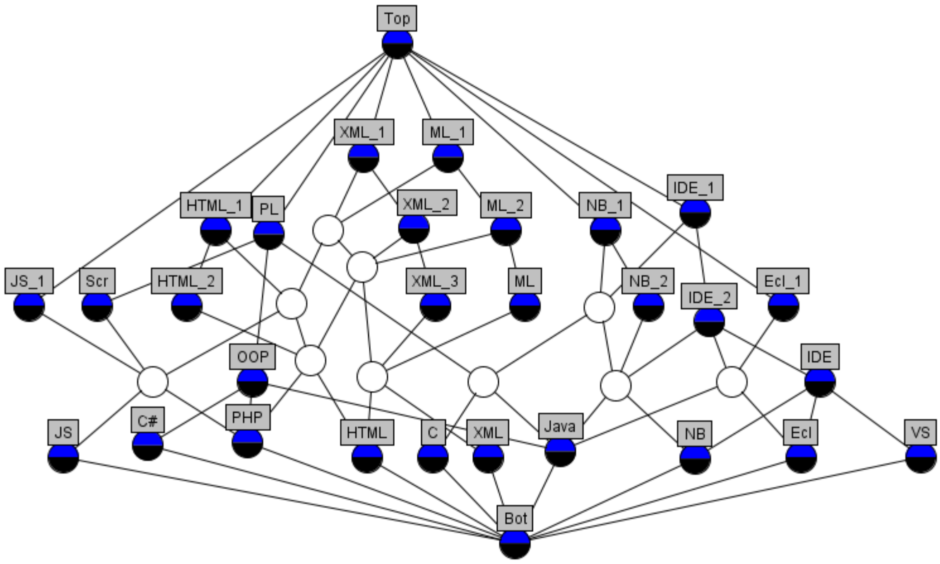

Figure 4: The concept lattice corresponding to the ontology lattice with added extra edges of Figure 1.

Theorem 3. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges and $cl((\mathcal{L}, E_{ext})) = (\mathcal{B}(V_1', V_2', T'), \leq)$ be the concept lattice constructed from G and w_{fil} be the filter weighting function. Let $O \subseteq S$ be an offer and $A \subseteq S$ be an application. Then,*

$$match_{sym}(A, O) = \frac{w_{fil}(F_A \cap F_O)}{w_{fil}(F_O)}. \tag{6}$$

The case of fuzzy offers and applications has the same complication as was with the extension graph. Similarly, we can salvage by charging the contributions of named concepts to the "top one with the same name". That is, we define $t(x, v) = \sum_{\langle I_x, J_x\rangle \leq \langle I_{v_i}, J_{v_i}\rangle \leq \langle I_{v_1}, J_{v_1}\rangle} w_{con}(\langle I_{v_i}, J_{v_i}\rangle)$ if $\langle I_x, J_x\rangle \leq \langle I_{v_1}, J_{v_1}\rangle$ and 0 otherwise, for all pairs $x, v \in S$. This again, is a preprocessing step. For $X \in \mathbf{F}(S)$ let $w_X^f(\langle I_{v_1}, J_{v_1}\rangle) = \{\langle I_{v_1}, J_{v_1}\rangle = \max_{x \in \mathsf{supp}(X)} \mu_X(x) t(x, v)\}$ and $w_X^f(\langle I_{v_i}, J_{v_i}\rangle) = 0$ for $i > 1$, as well as $w_X^f(\langle I, J\rangle) = 0$ if $I \cap J = \emptyset$. Furthermore, for the filter F_X of the concept lattice $\mathcal{B}(V_1', V_2', T'))$ generated by $\mathsf{supp}(X)$ let $fuzz_{fil}(F_X) = \{\langle I, J\rangle\colon w_X^f(\langle I, J\rangle) | \langle I, J\rangle \in F_X\}$ be a fuzzy set. Then the following can be proven along the lines of the proof of Theorem 2.

Theorem 4. *For a given offer $O \in \mathbf{F}(S)$ and application $A \in \mathbf{F}(S)$ we have*

$$match_{sym}(A, O) = \frac{||fuzz_{fil}(F_A) \cap fuzz_{fil}(F_O)||}{||fuzz_{fil}(F_O)||}. \tag{7}$$

3.3 Extremal problems

It is a natural question how the size of the original ontology lattice $\mathcal{L} = (S, \preceq)$ relates to the sizes of the extension graph $ext(\mathcal{L}, E_{ext})$ and the concept lattice $cl((\mathcal{L}, E_{ext}))$ obtained from $ext((\mathcal{L}, E_{ext}))$.

The proofs of the following statements can be found in the conference paper [28]. First, let us consider $ext(\mathcal{L}, E_{ext})$.

Proposition 1. *Let $\mathcal{L} = (S, \preceq)$ be an ontology lattice of $n+2$ nodes. Then for $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ we have $|V'| \leq n^2 + 2$. Furthermore, this estimate is sharp, that is for every positive integer n there exists ontology $\mathcal{L}_n = (S_n, \preceq)$ and set of extra edges E_{ext} such that $ext(\mathcal{L}_n, E_{ext})$ has $n^2 + 2$ vertices.*

The extremal example is shown on Figure 5.

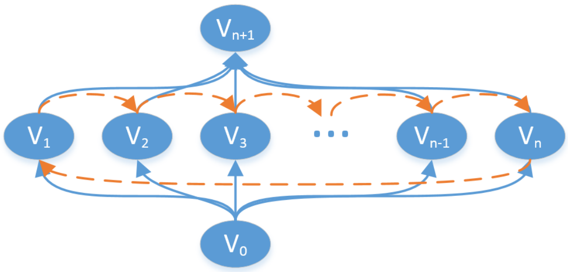

Figure 5: Extremal example

Our next goal is to bound the size of concept lattice $cl((\mathcal{L}, E_{ext}))$. The main question is how many "dummy" vertices are generated, that is concepts $\langle I, J \rangle$ such that $I \cap J = \emptyset$.

Theorem 5. *Let $\mathcal{L} = (S, \preceq)$ be an ontology lattice of $n+2$ nodes. Then for a set E_{ext} of extra edges $|cl((\mathcal{L}, E_{ext}))| \leq 2^n + n^2 - n + 1$ and this estimate is sharp, that is there exist $\mathcal{L}_n = (S_n, \preceq)$ and and set of extra edges E_{ext} such that $|cl((\mathcal{L}_n, E_{ext}))| = 2^n + n^2 - n + 1$.*

We have the same extremal example shown on Figure 5 as before.

Another interesting question could be how the average or expected size of extension graph and the concept lattice relates to the size of the original ontology lattice. This is the topic of further investigations. The first task is finding a reasonable probability distribution for the extra edges.

4 Strict Approach

As it was mentioned above, extra edges can be used based on different philosophies when extending offers. In this section we investigate how strict matching values of applications to offers can be preserved in the extension graph and in the concept lattice.

4.1 Preserving Strict Matching for crisp offers and applications

The main problem of preserving strict matching values in the extension graph is if extra edges are used to extend the offer, then extra nodes might appear in the extended offer whose weights are greater then 0. However, to address this problem, special node weighting functions can be defined depending on the offers.

For an offer $O \subset S$ let w^O_{node} be a node weighting function that preserves the weights of the nodes that are available from O via lattice edges in G, and the nodes that were generated from such nodes in G', and it assigns 0 to the other nodes, i.e. for a node $v \in V'$ let

$$w^O_{node}(v) = \begin{cases} w_{node}(v) & \text{if } \exists v_j \in ext_{E_{lat}}(O)\colon v \in V_j, \\ 0 & \text{otherwise.} \end{cases}$$

For $X \subset S$ let $ext^O_{E'}(X) = \{v : w^O_{node}(v) | \exists x \in X \text{ such that } p_{E'}(x, v) \neq \emptyset\}$ Note, that computing w^O_{node} is a preprocessing step that has to be done once for all offers, and then w^O_{node} can be reused to calculate matching values of applications to the given offer.

With these weighting function a similar result can be shown as in Lemma 1.

Lemma 3. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph, Let $O \subseteq S$ be an offer with w^O_{node} and let $A \subseteq S$ be an application. Then,*

$$match(A, O) = \frac{||ext_E(A) \cap ext_{E_{lat}}(O)||}{||ext_{E_{lat}}(O)||} = \frac{||ext_{E'}(A) \cap ext^O_{E'}(O)||}{||ext_{E'}(O)||} \tag{8}$$

Proof. The proof is analogous to Lemma 1's. However, $ext_{E'}(A) \cap ext_{E'}(O)$ may contain chain fragment $(v_{y_k}, \ldots, v_{y_1})$ of a chain $q_y = \{v_{y_l}, \ldots, v_{y_1}, top\}$ with base node v_y where v_y is only available from O via extra edges in G, i.e. $v_y \in ext_E(O) \setminus ext_{E_{lat}}(O)$. But w^O_{node} assigns 0 to such $v_{y_k}, \ldots, v_{y_1}$ nodes by definition. Therefore $||ext_{E'}(A) \cap ext^O_{E'}(O)|| = \sum_{u \in ext_{E'}(A) \cap ext_{E'}(O)} \min(w_{node}(u), w^O_{node}(u)) = ||ext_E(A) \cap ext_{E_{lat}}(O)||$ and analogously, $||ext_{E_{lat}}(O)|| = ||ext^O_{E'}(O)||$. Thus equation (8) holds as well. □

The same issue appears if we want to preserve strict matching values of crisp applications to crisp offers in the concept lattice as we solved in case of the extension graph, namely extended offer might contain new nodes with weight greater than 0. However, the offer specific weighting functions solve this issue as well.

We extend w^O_{node} for concepts, namely let w^O_{con} be a concept weighting function generated by an offer O such that for a concept $\langle I, J \rangle$:

$$w^O_{con}(\langle I, J \rangle) = \begin{cases} w_{con}(\langle I, J \rangle) & \text{if } I \cap J = \{v\} \text{ such that } \exists v_j \in ext_{E_{lat}}(O)\colon v \in V_j, \\ 0 & \text{otherwise.} \end{cases}$$

Let w^O_{fil} be the filter weighting function based on w^O_{con}, i.e for a filter $F \in \mathcal{P}(\mathcal{B}(V_1', V_2', T'))$: $w^O_{fil}(F) = \sum_{\langle I,J\rangle \in F} w^O_{con}(\langle I, J\rangle)$.

With these weighting functions, we can prove the following theorem similarly to Theorem 3.

Theorem 6. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges and $cl((\mathcal{L}, E_{ext})) = (\mathcal{B}(V_1', V_2', T'), \leq)$ be the concept lattice constructed from G and w_{fil} be the filter weighting function. Let $O \subseteq S$ be an offer with w^O_{con} and w^O_{fil} concept and filter weighting functions, respectively and let $A \subseteq S$ be an application. Then,*

$$match(A, O) = \frac{w^O_{fil}(F_A \cap F_O)}{w^O_{fil}(F_O)} \tag{9}$$

Proof. Analogously to Theorem 3's proof and based on Lemma 1 it is enough to prove that

$$\frac{w^O_{fil}(F_A \cap F_O)}{w^O_{fil}(F_O)} = \frac{||ext_{E'}(A) \cap ext^O_{E'}(O)}{||ext^O_{E'}(O)||}. \tag{10}$$

However, F_A and F_O contain concepts for all nodes of $ext_{E'}(A)$ and $ext_{E'}(O)$ respectively. But w^O_{con} assigns 0 to such $\langle I_v, J_v\rangle$ concepts where $v \in V'$ is not contained in any chain whose base was available from O in G using lattice edges only. Therefore w^O_{fil} sums up the same values as w^O_{fset}, i.e. equation (10) holds as well. □

4.2 Strict matching for fuzzy offers and applications

If offers and applications are allowed to be fuzzy sets, that is $O, A \in \mathbf{F}(S)$, then we are confonted with the same problem as we saw in the symmetric case. Consider the lattice and extension graph of Figure 2. If $O = \{D{:}\,0.6\}$ and $A = \{B{:}\,0.9\}$, then $ext_{E_{lat}}(O) = \{D{:}\,0.6, top{:}\,0.6\}$ and $ext_E(A) = \{A{:}\,0.9, B{:}\,0.9, D{:}\,0.54, C{:}\,0.432, top{:}\,0.9\}$, so $ext_E(A) \cap ext_{E_{lat}}(O) = \{D{:}\,0.54, top{:}\,0.6\}$. If again we apply definition for the extension graph mechanically we would get $ext'_{E'}(X) = \{v{:}\,\gamma_v | \gamma_v = \max \mu_X(x) w_{node}(v)$ for $x \in \mathsf{supp}(X)$ and $p_{E'}(x, v) \neq \emptyset\}$ for applications and $ext'_{E'}(X) = \{v{:}\,\gamma_v | \gamma_v = \max \mu_X(x) w^O_{node}(v)$ for $x \in \mathsf{supp}(X)$ and $p_{E'}(x, v) \neq \emptyset\}$. However, this definition would give

$$ext'_{E'}(O) = \{D{:}\,0.18, D_2{:}\,0.06, D_1{:}\,0.36, top{:}\,0.6\}$$

and

$$ext'_{E'}(A) = \{A{:}\,0.18, A_1{:}\,0.72, B{:}\,0.9, C_1{:}\,0.432, D_1{:}\,0.54, top{:}\,0.9\}$$

resulting in

$$ext'_{E'}(A) \cap ext'_{E'}(O) = \{D_1{:}\,0.36, top{:}\,0.6\}.$$

Thus, $||ext'_{E'}(A) \cap ext'_{E'}(O)|| \neq ||ext_E(A) \cap ext_E(O)||$.

To avoid this anomaly we again charge the contributions of node weights to the top elements of chains, as in the symmetric case. Recall that for $x, v \in S$

$$t(x,v) = \sum_{i=1}^{m} w_{node}(v_i), \tag{11}$$

where (x, v_m) is the edge of the extension graph G' from x to the chain q_v. If no such edge exists then $t(x,v)$ is defined to be 0. Also for $X \in \mathbf{F}(S)$, $ext^{f}_{E'}(X) = \{v_1 \colon \gamma_{v_1} | v \in S \text{ and } \gamma_{v_1} = \max_{x \in \mathsf{supp}(X)} \mu_X(x) t(x,v)\}$ was introduced. Now, let $x, v \in S$. Define $t^O(x,v)$ by replacing $w_{node}(v_i)$ by $w^O_{node}(v_i)$ in (11). Furthermore for $X \in \mathbf{F}(S)$, let $ext^{fO}_{E'}(X) = \{v_1 \colon \gamma_{v_1} | v \in S \text{ and } \gamma_{v_1} = \max_{x \in \mathsf{supp}(X)} \mu_X(x)$ $t^O(x,v)\}$. The proof of the following theorem is straightforward analogue of that of Theorem 2

Theorem 7. *Let $G = (V, E)$ be a directed graph extending the lattice $\mathcal{L} = (S, \preceq)$ with extra edges, $G' = ext(\mathcal{L}, E_{ext}) = (V', E')$ be the extension graph. Let $O \in \mathbf{F}(S)$ be an offer and $A \in \mathbf{F}(S)$ be an application . Then*

$$match(A,O) = \frac{||ext_E(A) \cap ext_{E_{lat}}(O)||}{||ext_{E_{lat}}(O)||} = \frac{||ext^f_{E'}(A) \cap ext^{fO} f_{E'}(O)||}{||ext^{fO}_{E'}(O)||}. \tag{12}$$

5 Related Work

The aim of profile matching is to find the most fitting candidates to given profiles. Due to its various applications areas, it has become a widely investigated topic recently. Profiles can be represented as sets of elements and then numerous set similarity measures [3], such as Jaccard or Sørensen-Dice, are applicable to compute matching values.

There exist methods assuming that elements of profiles are organized into a hierarchy or ontology. For example, Lau and Sure [16] proposed an ontology-based skill management system for eliciting employee skills and searching for experts within an insurance company. Ragone et al. [29] investigated peer-to-peer e-market place of used cars and presented a fuzzy extension of Datalog to match sellers and buyers based on required and offered properties of cars. Di Noia et al. [5] placed matchmaking on a consistent theoretical foundation using description logic. They defined matchmaking as information retrieval task where demands and supplies are expressed using the same semi-structured data in form of advertisement and task results are ranked lists of those supplies best fulfilling the demands.

Guedj [11] claims that applying semantic matching technologies has the problem that requesting to the user to weighing the skills is a barrier to an usability and an efficiency of such methods on the user point of view and propses a first approach to solve this problem. Tinelli et.al [33] combine the representation power of a logical language with the information processing efficiency of a DBMS and implement it in the platform I.M.P.A.K.T. Shen et.al. [32] use AI to jointly model job description, candidate resume and interview assessment.Yan et.al. [36] realize that interviewers and job seekers have preferences and propose to learn job-resume matching methods with the hidden preference information incorporated. Pitukhin et.al. [25] take "one sided" question: they present methods to gather and rank job offers from the point of view of the applicant, starting from the assumption that there are many offers that could not be properly assessed by hand.

With respect to foundations of a profile matching theory the first promising attempt to take hierarchical dependencies into account was done by Popov and Jebelean [26], which defines the initial filter-based measure. However, weights are not used, only cardinalities, which correspond to the special case that all concepts are equally weighted. The matching theory in [21] is inspired by this work, but takes the filter-based approach much further. To our best knowledge no other approach in this direction has been tried, though sophisticated taxonomies in the recruitment domain such as DISCO [6], ISCO [14] and ISCED [13] already exist. Ontologies have also been used in the area of recruiting in connection with profile matching (see [7] for a survey). However, while it is claimed that matching accuracy can be improved [23], the matching approach itself remains restricted to Boolean matching, which basically means to count how many requested skills also appear in a given profile [22].

In [27] an extension to the matching theory has been proposed, which exploits relations between the concepts in a profile that are not covered by the lattice, i.e. the presence of a particular concept in a profile may only partially imply the presence of another concept. Such additional links between the elements of the lattice may be associated with a degree (or probability) and even cycles may be permitted. This leads to an *enriched matching theory* by means of values associated to paths, which enables an interpretation using fuzzy filters [12]. For the probabilistic extension this research exploits probabilistic logic with maximum entropy semantics in [1, 15], for which sophisticated reasoning methods exist [30]. In the meantime this research has been taken further showing that it is possible to compute an extended lattice such that matching measures for profiles in the extended lattice capture exactly the same as the path values [28].

We also assumed a structure among elements of profiles that can be represented by an ontology, which then fulfills lattice properties, so profiles can be represented as filters. However, we extended the ontology lattice with extra edges to capture such relationships that subsumptions cannot express. Then we showed how these edges are usable to refine the ontology.

There are several methodologies to learn ontologies from unstructured texts or semi-structured data [2, 31]. Besides identifying concepts, discovering relationships between the concepts is a crucial part of ontology construction and refinement. Text-To-Onto [20] uses statistical, data mining, and pattern-based approaches over text corpus to extract taxonomic and non-taxonomic relations. In [34], various similarity measures were introduced between semi-structured Wikipedia infoboxes and then SVMs and Markov Logic Networks were used to detect subsumptions between infobox-classes.

We presented a method to refine an ontology based on extra edges that represent some sort of quantifiable relationship between concepts in a profile. These relationships can be given by domain experts, computed from statistics, or result from data mining techniques. For example, in [37] the authors used association rules and latent semantic indexing over job offers to detect relationships between competencies. In our method we defined profile extensions and weighting functions as well to preserve matching values of profiles computed from edge weights.

Formal concept analysis (FCA) [9] is also used to build and maintain formal ontologies. For example, Cimiano et al. [4] presented a method of automatic acquisition of concept hierarchies from a text corpus based on FCA. In [19], the authors used FCA to revise ontology when new knowledge was added to it. In our method we used FCA to restore lattice properties after added new nodes and edges to it based on extra edges. However as we focused on preserving matching values of profiles during the transformations, we adapted our profile weighting functions to the modified ontology lattice as well.

6 Conclusions

In this article the approach of [28] was extended to fuzzy sets of offers and applications. We refined the matching theory with profiles represented by filters in a lattice. Such a lattice can be obtained from a knowledge base as shown in [24]. The basis for the theory is the definition of weighted matching measures on pairs of such filters. For instance, in the field of human resources profiles correspond to skills sets of job applicants as well as to requirements in job offers. Learning matching weight from human expertise as well as efficient querying have been handled in [21]. We now investigated how ontology lattices can be extended by additional information and used for matching. We defined matching functions to find the most suitable applicant to a job offer, however, our results are applicable in other fields as well.

First, profiles are represented as filters in an ontology lattices, which capture subsumption relations between concepts. Then, we extend such an ontology lattice by additional information in the form of extra edges describing additional quantifiable relations between the concepts. A directed graph is built from the lattice endowed with extra edges to handle directed cycles that the new edges might have introduced, and matching functions are defined based on reachable nodes from the nodes in a profiles.

Two approaches were presented to extend profiles with derived nodes. In the first one, both the given and the requested profiles were extended, as profiles should be handled uniformly. In the second approach, only the given profiles were extended, which helps to distinguish cases, where the given requirements are met directly from those, where the requirements are only met by the combination of several concepts that all contribute partially to the requirements. For instance, in the human resources field the second strict approach may help employers to better differentiate among job applicants.

We presented a method that eliminates directed cycles from the graph. It constructs an extension graph by adding node chains to the original lattice based on directed paths between nodes in the directed graph and node weights got also modified as part of the construction. The extension graph is a directed acyclic graph and therefore a poset but it is not necessary a lattice. We further exploited formal concept analysis to extend the poset to a concept lattice so that filters of this lattice could be used to calculate matching values. Different node weightings were used to preserve the original matching values in the two approaches. Comparisons of the sizes of the ontology lattice and the generated acyclic directed graph, as well as the concept lattice were also given.

This shows that the matching theory from [21] is rather powerful, as it captures de facto the fuzzy extensions.

References

[1] Beierle, Christoph, Finthammer, Marc, and Kern-Isberner, Gabriele. Relational probabilistic conditionals and their instantiations under maximum entropy semantics for first-order knowledge bases. *Entropy*, 17(2):852–865, 2015. DOI: 10.3390/e17020852.

[2] Buitelaar, Paul, Cimiano, Philipp, and Magnini, Bernardo. Ontology learning from text: An overview. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*, Volume 123 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press, Amsterdam, 2005. URL: https://pub.uni-bielefeld.de/record/2497696.

[3] Choi, Seung-Seok, Cha, Sung-Hyuk, and Tappert, Charles C. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[4] Cimiano, Philipp, Hotho, Andreas, and Staab, Steffen. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24(1):305–339, 2005. DOI: 10.1613/jair.1648.

[5] Di Noia, Tommaso, Di Sciascio, Eugenio, and Donini, Francesco M. Semantic matchmaking as non-monotonic reasoning: A description logic approach. *Journal of Artificial Intelligence Research*, 29:269–307, 2007. DOI: 10.1613/jair.2153.

[6] European dictionary of skills and competences. URL: http://www.disco-tools.eu.

[7] Falk, Thorsten et al. Semantic-Web-Technologien in der Arbeitsplatzvermittlung. *Informatik Spektrum*, 29(3):201–209, 2006. DOI: 10.1007/s00287-006-0061-4.

[8] Ganter, Bernhard and Kuznetsov, Sergei O. Stepwise construction of the Dedekind-MacNeille completion. In Mugnier, Marie-Laure and Chein, Michel,

editors, *Conceptual Structures: Theory, Tools and Applications*, pages 295–302, Berlin, Heidelberg, 1998. Springer. DOI: 10.1007/BFb0054922.

[9] Ganter, Bernhard, Stumme, Gerd, and Wille, Rudolf. Formal concept analysis: Theory and applications. *Journal of Universal Computer Science*, 10(8):926, 2004. DOI: 10.3217/jucs-010-08.

[10] Ganter, Bernhard and Wille, Rudolf. *Formal Concept Analysis: Mathematical Foundations*. Springer Science & Business Media, 2012.

[11] Guedj, Michaël. Levelized taxonomy approach for the job seeking/recruitment problem. In *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) and 15th International Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pages 448–451, 2016. DOI: 10.1109/CSE-EUC-DCABES.2016.222.

[12] Hájek, Petr. *Mathematics of Fuzzy Logic*. Kluwer Academic Publishers, 1998.

[13] International standard classification of education. URL: http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx.

[14] International standard classification of occupations. URL: http://www.ilo.org/public/english/bureau/stat/isco/isco08/, 2008.

[15] Kern-Isberner, Gabriele and Lukasiewicz, Thomas. Combining probabilistic logic programming with the power of maximum entropy. *Artificial Intelligence*, 157(1-2):139–202, 2004. DOI: 10.1016/j.artint.2004.04.003.

[16] Lau, Thorsten and Sure-Vetter, York. Introducing ontology-based skills management at a large insurance company. In *Modellierung (Proceedings)*, pages 123–134, 2002.

[17] Levandowsky, Michael and Winter, David. Distance between sets. *Nature*, 234(5):34–35, 1971. DOI: 10.1038/234034a0.

[18] Liu, Lianzhen and Li, Kaitai. Fuzzy filters of BL-algebras. *Information Sciences*, 173(1):141–154, 2005. DOI: 10.1016/j.ins.2004.07.009.

[19] Looser, Dominic, Ma, Hui, and Schewe, Klaus-Dieter. Using formal concept analysis for ontology maintenance in human resource recruitment. In Ferrarotti, Flavio and Grossmann, Georg, editors, *Proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling*, Volume 143, pages 61–68. Australian Computer Society, 2013. DOI: 10.5555/2527198.2527204.

[20] Maedche, Alexander and Volz, Raphael. The ontology extraction & maintenance framework Text-To-Onto. In *Proceedings of the Workshop on Integrating Data Mining and Knowledge Management*, pages 1–12, 2001.

[21] Martínez Gil, Jorge, Paoletti, Alejandra Lorena, Rácz, Gábor, Sali, Attila, and Schewe, Klaus-Dieter. Accurate and efficient profile matching in knowledge bases. *Data & Knowledge Engineering*, 117:195–215, 2018. DOI: 10.1016/j.datak.2018.07.010.

[22] Mochol, Malgorzata, Nixon, Lyndon J. B., and Wache, Holger. Improving the recruitment process through ontology-based querying. In Bontas Simperl, Elena Paslaru, Hepp, Martin, and Tempich, Christoph, editors, *Proceedings of the First International Workshop on Applications and Business Aspects of the Semantic Web*, Volume 226 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006. URL: https://ceur-ws.org/Vol-226/paper05.pdf.

[23] Mochol, Malgorzata, Wache, Holger, and Nixon, Lyndon J. B. Improving the accuracy of job search with semantic techniques. In Abramowicz, Witold, editor, *Proceedings of the 10th International Conference on Business Information Systems*, Volume 4439 of *Lecture Notes in Computer Science*, pages 301–313. Springer, 2007. DOI: 10.1007/978-3-540-72035-5_23.

[24] Paoletti, Alejandra Lorena, Martinez-Gil, Jorge, and Schewe, Klaus-Dieter. Extending knowledge-based profile matching in the human resources domain. In Chen, Qiming et al., editors, *Database and Expert Systems Applications (DEXA 2015) Part II*, Volume 9262 of *Lecture Notes in Computer Science*, pages 21–35. Springer, 2015. DOI: 10.1007/978-3-319-22852-5_3.

[25] Pitukhin, E., Astafyeva, M., and Astafyeva, I. Methodology for job advertisements analysis in the labor market in metropolitan cities: The case study of the capital of Russia. In Silhavy, R., editor, *Intelligent Algorithms in Software Engineering*, Volume 1224 of *Advances in Intelligent Systems and Computing*. Springer, 2020. DOI: 10.1007/978-3-030-51965-0_37.

[26] Popov, Nikolaj and Jebelean, Tudor. Semantic matching for job search engines – a logical approach. Technical Report 13-02, Research Institute for Symbolic Computation, JKU Linz, 2013.

[27] Rácz, Gábor, Sali, Attila, and Schewe, Klaus-Dieter. Semantic matching strategies for job recruitment: A comparison of new and known approaches. In Gyssens, Marc and Simari, Guillermo Ricardo, editors, *Foundations of Information and Knowledge Systems (FoIKS 2016)*, Volume 9616 of *Lecture Notes in Computer Science*, pages 149–168. Springer, 2016. DOI: 10.1007/978-3-319-30024-5_9.

[28] Rácz, Gábor, Sali, Attila, and Schewe, Klaus-Dieter. Refining semantic matching for job recruitment: An application of formal concept analysis. In Ferrarotti, Flavio and Woltran, Stefan, editors, *Foundations of Information and Knowledge Systems (FoIKS 2018)*, Volume 10833 of *Lecture Notes in Computer Science*, pages 322–339, 2018. DOI: 10.1007/978-3-319-90050-6_18.

[29] Ragone, Azzurra, Straccia, Umberto, Di Noia, Tommaso, Di Sciascio, Eugenio, and Donini, Francesco M. Fuzzy matchmaking in e-Marketplaces of Peer Entities using Datalog. *Fuzzy Sets and Systems*, 160(2):251–268, 2009. DOI: 10.1016/j.fss.2008.07.002.

[30] Schramm, Manfred and Ertel, Wolfgang. Reasoning with probabilities and maximum entropy: The system PIT and its application in LEXMED. In *Operations Research Proceedings 1999*. Springer, 2000. DOI: 10.1007/978-3-642-58300-1_41.

[31] Shamsfard, Mehrnoush and Barforoush, Ahmad Abdollahzadeh. The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(4):293–316, 2003. DOI: 10.1017/S0269888903000687.

[32] Shen, Dazhong, Zhu, Hengshu, Zhu, Chen, Xu, Tong, Ma, Chao, and Xiong, Hui. A joint learning approach to intelligent job interview assessment. In Lang, Jérôme, editor, *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3542–3548, 2018. DOI: 10.24963/ijcai.2018/492.

[33] Tinelli, Eufemia, Colucci, Simona, Donini, Francesco M., Di Sciascio, Eugenio, and Giannini, Silvia. Embedding semantics in human resources management automation via SQL. *Applied Intelligence*, 46:952–982, 2017. DOI: 10.1007/s10489-016-0868-x.

[34] Wu, Fei and Weld, Daniel S. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the 17th International Conference on World Wide Web*, pages 635–644. ACM, 2008. DOI: 10.1145/1367497.1367583.

[35] Wygralak, Maciej. *Cardinalities of Fuzzy Sets*. Springer, 2003. DOI: 10.1007/978-3-540-36382-8.

[36] Yan, Rui, Le, Ran, Song, Yang, Zhang, Tao, Zhang, Xiangliang, and Zhao, Dongyan. Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922. Association for Computing Machinery New York NY, 2019. DOI: 10.1145/3292500.3330963.

[37] Ziebarth, Sabrina, Malzahn, Nils, and Hoppe, Heinz Ulrich. Using data mining techniques to support the creation of competence ontologies. In *Frontiers in Artificial Intelligence and Applications*, pages 223–230, 2009. DOI: 10.3233/978-1-60750-028-5-223.

Received 11th August 2021