An EBMC-Based Approach to Selecting Types for Entity Filtering

Jiwei Ding, Wentao Ding, Wei Hu and Yuzhong Qu
National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{jiweiding,wentaoding}@outlook.com, {whu,yzqu}@nju.edu.cn

Abstract

The quantity of entities in the Linked Data is increasing rapidly. For entity search and browsing systems, filtering is very useful for users to find entities that they are interested in. Type is a kind of widely-used facet and can be easily obtained from knowledge bases, which enables to create filters by selecting at most K types of an entity collection. However, existing approaches often fail to select high-quality type filters due to complex overlap between types. In this paper, we propose a novel type selection approach based upon Budgeted Maximum Coverage (BMC), which can achieve integral optimization for the coverage quality of type filters. Furthermore, we define a new optimization problem called Extended Budgeted Maximum Coverage (EBMC) and propose an EBMC-based approach, which enhances the BMC-based approach by incorporating the relevance between entities and types, so as to create sensible type filters. Our experimental results show that the EBMC-based approach performs best comparing with several representative approaches.

Introduction

Along with the rapid growth of the Linked Data, many Web search and browsing activities are centered on entities. General users often use entity search or browsing systems, such as Libra Academic Search (Nie et al. 2007) and Falcons (Cheng and Qu 2009), to look for entities that they are interested in. However, it might be difficult for users to find the entities they want when search results contain large amount of entities. Therefore, entity filtering is demanded by users to ease their tasks.

Faceted search systems provide filters in different facets for interactive search and browsing (Sah and Wade 2013). Since type can be seen as an important facet (Tonon et al. 2013), a group of filters can be created by selecting values in this facet. In many publicly-accessible knowledge bases such as DBpedia and Yago, entities are associated with multiple types. The number of types related to the whole entity collection can be very large, while the number of filters in a group is limited to a positive integer K. How to select at most K types to make up a group of filters in entity search or browsing systems is worth studying.

Let us take an answer set to the question *"chess players who died in the same place they were born in"* as an example. The resulted entity collection consists of 45 chess players, each of which has at least 13 Yago types.

To the best of our knowledge, the most commonly used approach in faceted search is to count values in terms of their occurrence frequency, which means to rank types by their coverage rate in our problem. The following list shows a group of filters generated by this approach.

ChessPlayer (44)
Contestant (44)
Player (44)
Rival (19)
Intellectual (18)

However, we argue that this group of filters are not good enough. Although it covers most of entities in the collection, some types like *"Player"* is not suitable for filtering, since it covers too many entities. Furthermore, the overlap between types is not considered in this approach, e.g., *"Contestant"* and *"Player"*, which is another main cause to this drawback.

Some approaches are proposed to reduce the overlap between selected types, such as Maximum Marginal Relevance (MMR) (Carbonell and Goldstein 1998), but they still cannot avoid selecting types like *"ChessPlayer"*, which covers almost all of the entities. Some other approaches (Hearst 2006; Cheng and Qu 2009) also provide their solutions, but all of them suffer from a serious drawback, that is, they cannot provide sensible types, in other words, users cannot decide how close in perception the entity they want is relevant to a type. For example, the filter *"ChessOlympiadCompetitors (16)"* is more sensible than *"Intellectual (18)"*.

In this paper, we use the Budgeted Maximum Coverage (BMC) model (Khuller, Moss, and Naor 1999) to achieve an integral optimization for the coverage quality of selected types, which increases the coverage rate and reduces the overlap at the same time. To decrease the influence of the selected types covering too many or too few entities, we design a cost function on candidate types based on the coverage rates of types. Furthermore, we define the Extended Budgeted Maximum Coverage (EBMC) optimization problem and propose an EBMC-based approach, which enhances the BMC-based approach by incorporating the relevance between entities and types. In our implemented method, we leverage the type hierarchy to measure the relevance of an

entity with its associated types. Our experimental results show that the EBMC-based approach performs best in overall comparing with other methods. For the "chess players" example, the type filters selected by the EBMC-based approach are shown below. This group of types is much more sensible than the previous one.

ChessOlympiadCompetitors (16)
ChessTheoreticians (15)
SovietChessPlayers (14)
GermanChessPlayers (4)
EnglishChessPlayers (5)

The rest of this paper is structured as follows. Section 2 reviews related work. In Section 3, we propose an approach based on BMC. In Section 4, we define EBMC and propose an approximation algorithm to solve it. Also, an EBMC-based approach is proposed. In Section 5, we evaluate the two proposed approaches by comparing them with three others on various entity collections. Finally, Section 6 concludes the paper with future work.

Related Work

Our work is related to faceted search since type is often considered as an important kind of facets. To the best of our knowledge, the most commonly used approach for selecting values in a facet is based on occurrence frequency. However, this approach is not suitable for selecting types since there often exists high overlap between different types.

MMR (Carbonell and Goldstein 1998) is a diversity-based re-ranking approach, which is mainly used for re-ranking search results or summarizing documents. We may use this approach here to minimize overlap and increase the coverage at the same time. However, this approach still does not work well because it cannot avoid types that covers too generally or too specifically.

Falcons Object Search (Cheng and Qu 2009) is a search engine for objects in the Linked Data. It gives a simple solution to this problem, which selects types according to coverage rate mainly, and ensures that class inclusion relation never holds between any pair of selected types. However, this approach may lead to a low coverage percentage when the number of selected types is limited to K, and sometimes suffers from providing inadequate types.

An alternative solution to this problem is "hierarchical faceted category", since types form a hierarchy naturally. However, the type hierarchy is sometimes too detailed, which means it may cost users too many clicks before they get the filtering condition they want. In the "chess player" example, at least 4 clicks are needed between the type *"Player"* and a specific type *"EnglishChessPlayers"*.

Another line of related work is entity type ranking. TRank (Tonon et al. 2013) proposes different kinds of ranking approaches and evaluates the efficiency, but these approaches cannot be applied to entity collection directly.

Besides, our work is also related to tag recommendation if we broadly treat types as tags. Xu et al. exploit collaborative tagging information to recommend tags (Xu et al. 2006). Their recommendation algorithm attempts to minimize the overlap among the recommended tags to allow for high coverage of multiple facets. However, this algorithm can only be used for a single entity and needs much user tagging information.

Selecting Types Based on BMC

In this section, we firstly introduce the well-known BMC optimization problem and its approximation algorithm. Then, we transform our problem to the BMC form, and mainly focus on how to assign costs to different types.

Budgeted Maximum Coverage Problem

Notice that our problem can be seen as selecting at most K types to cover most of the entities in the collection, we can formalize our type selection problem by the BMC optimization problem (Khuller, Moss, and Naor 1999), which is defined as follows:

Definition 1 (The Budgeted Maximum Coverage Problem)**.** *A collection of sets $S = \{S_1, S_2, \ldots, S_m\}$ with associated costs $\{c_i\}_{i=1}^m$ is defined over a domain of elements $X = \{x_1, x_2, \ldots, x_n\}$ with associated weights $\{w_i\}_{i=1}^n$. The goal is to find a collection of sets $S' \subseteq S$, such that the total cost of elements in S' does not exceed a given budget L, and the total weight of elements covered by S' is maximized.*

Although BMC is an NP-hard problem, several efficient approximation algorithms have been developed. By comparing the approximation ratio and the time complexity of these algorithms, we consider the $(1 - \frac{1}{\sqrt{e}})$ approximation algorithm with time complexity $O(m^2 n)$ provided by (Khuller, Moss, and Naor 1999) to be best for our work.

The BMC-Based Approach

To formalize our problem, entities can be regarded as elements in BMC. Set S_i is defined as a set of entities that are associated with $type_i$, and we may simply represent this set by $type_i$. Without loss of generality, we set budget L to 1 to simplify the formalization. The weight of each entity can be set by its popularity, its relevance to the query, or simply 1 so that all entities are treated equally.

As for the cost of a type, it should be set by how appropriate it is to be selected as a filter, which can be inferred from the coverage rate of the type filter. In our problem, a good group of filters consists at most K types with similar coverage rate and little overlap between each other. A type with huge coverage rate may be too vague and leads to a large overlap with other selected types, while a type with little coverage rate may not contain the entity wanted by the users. So in the ideal case, the K type filters are orthogonal, each of which covers $\frac{1}{K}$ entities. Since the overlap between types cannot be avoided in practice, it is suggested that the ideal coverage rate R can be chosen between $\frac{1}{K}$ and $\frac{2}{K}$. A power function is considered to be the punishment for the types covering too many or too few entities. Lastly, we set $\frac{1}{K+1}$ as a lower bound of the cost for each type, which ensures that no more than K types would be selected by the approach. On the whole, the cost function can be defined as follows:

$$\left|\frac{|S_i|}{|S|} - R\right|^{\alpha} + \frac{1}{K+1}, \tag{1}$$

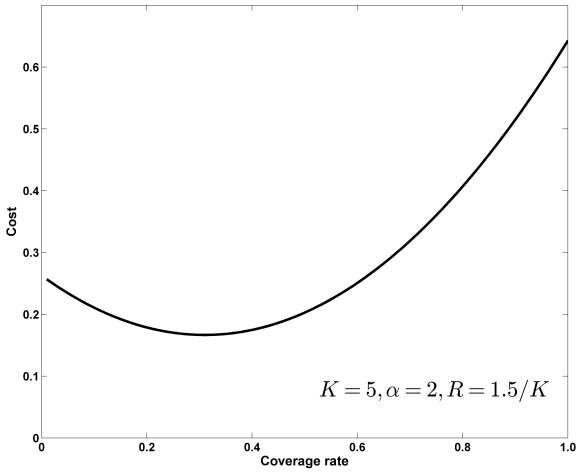

Figure 1: An example of cost function

where R is the ideal coverage rate for each type that can be set by application developers, and α is a parameter varying from different K or different datasets, which can be tuned by training. An example of cost function is shown in Figure 1.

Selecting Types Based on EBMC

In this section, we firstly analyze the drawback of the BMC-based approach. Then, we give a formal definition of the EBMC optimization problem and propose a $(1 - \frac{1}{\sqrt{e}})$-approximation algorithm. Finally, a type selection approach based on EBMC is presented.

Drawback of the BMC-Based Approach

By transforming the type selection problem to the BMC problem, we can decrease the effect of selecting types that cover too many or too few entities. However, this approach can still be improved since the binary type coverage function does not characterize how relevant a type is to an entity, e.g., the weight of covering *"VasilySmyslov"* by *"SovietChessPlayers"* is the same as covering it by *"Player"* in BMC. If these two types cover similar entities in an entity collection, the BMC-based approach would choose either of them. But the type *"SovietChessPlayers"* is more informative and also easy to understand, which means it is more suitable for being a filter. To realize this improvement, we generalize the set in BMC to fuzzy set, which leads to the EBMC problem.

Extended Budgeted Maximum Coverage Problem

Now, we define the EBMC problem as follows:

Definition 2 (The Extended Budgeted Maximum Coverage Problem)**.** *Let $S = \{S_1, S_2, \ldots, S_m\}$ be a collection of fuzzy sets defined over a domain of elements $X = \{x_1, x_2, \ldots, x_n\}$. Each fuzzy set has a cost $\{c_i\}_{i=1}^m$ while each element has a weight $\{w_i\}_{i=1}^n$. $f(S_i, x_j)$ is defined as the grade of membership of x_j in S_i. The goal is to find a collection of fuzzy sets $S' \subseteq S$, such that the total cost of S', denoted by $c(S')$, does not exceed a given budget L, while the total weight $w(S')$ is maximized.*

Where $c(S')$ and $w(S')$ are defined by the following formulae, respectively.

$$c(S') = \sum_{S_i \in S'} c_i \tag{2}$$

$$w(S') = \sum_{j=1}^{n} \big(w_j \cdot \max\{f(P, x_j) | P \in S'\}\big). \tag{3}$$

Obviously, the EBMC problem is a generalization of BMC. EBMC can also be treated as a specific case of the Generalized Maximum Coverage problem (GMC) (Cohen and Katzir 2008). However, the GMC problem is so complex that even the simplest algorithm mentioned in that paper has a high time complexity of $O(m^2 n \cdot f(n, \epsilon))$ and a non-constant approximation ratio of $\frac{e-1}{2e-1} - \epsilon$, where $O(f(n, \epsilon))$ is the time complexity of an FPTAS approximation algorithm of the knapsack problem of n items. Our evaluation with GMC shows that it does not work for real-time applications when n and m reach 100.

An Approximation Algorithm for the EBMC Problem

Following the idea of the approximation algorithm for solving BMC, we give an approximation algorithm for EBMC, as shown in Algorithm 1.

Algorithm 1: Approximation algorithm for EBMC

```
G ← ∅; C ← 0; U ← S
repeat
    Select S_i ∈ U that maximizes (w(G∪{S_i}) − w(G)) / c_i;
    if C + c_i ≤ L then
        G ← G ∪ {S_i};
        C ← C + c_i;
        U ← U \ {S_i};
until U = ∅;
Select a fuzzy set T that maximizes w({T}) over S;
if w(G) ≥ w({T}) then
    return G;
else
    return {T};
```

We give a brief result to show that the main properties of the BMC problem still holds after introducing the concept of fuzzy set, and the time complexity of Algorithm 1 is still $O(m^2 n)$. We proved that the algorithm is $(1 - \frac{1}{\sqrt{e}})$-approximate. The detailed version of proof is provided in Appendix A.

Here, we give two properties of the EBMC model, which form the basis of the proof.

Lemma 1. *For any two collections of fuzzy sets A and B:*

$$w(A) + w(B) \geq w(A \cup B). \tag{4}$$

Lemma 2. *For any two collections of fuzzy sets A and B:*

$$w(A) - w(B) \leq \sum_{P \in A \setminus B} (w(B \cup \{P\}) - w(B)). \quad (5)$$

Let l be the number of fuzzy sets in G, and S_{G_i} be the i^{th} fuzzy set added into G. $S_{G_{l+1}}$ is defined as the first fuzzy set from an optimal solution OPT selected but not added to G by the algorithm. In the following two lemmas we present some properties of the greedy approach.

Lemma 3. *After each iteration $i = 1, \ldots, l$,*

$$w(G_i) - w(G_{i-1}) \geq \frac{c(\{S_{G_i}\})}{L} \cdot \big(w(OPT) - w(G_{i-1})\big). \quad (6)$$

Lemma 4. *After each iteration $i = 1, \ldots, l$,*

$$w(G_i) \geq \Big(1 - \prod_{k=1}^{i} \big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT). \quad (7)$$

Now we can prove the approximation ratio and the time complexity of the algorithm.

Theorem 1. *Algorithm 1 achieves an approximation factor of $(1 - \frac{1}{\sqrt{e}}) \approx 0.393469$.*

Theorem 2. *The time complexity of Algorithm 1 is $O(m^2 n)$, where $m = |S|$ and $n = |X|$.*

The EBMC-Based Approach

The cost for types and the weight for entities can be set the same as in the BMC-based approach. By the modeling capability of EBMC, we can use the membership function to describe the relevance between the set and elements, which is different from the binary value in BMC.

To describe the relevance between types and entities, the hierarchy of types should be considered. Notice that the types and their hierarchy form a directed acyclic graph, which can be converted to a Hasse diagram since the hierarchy of types is a partially ordered relation. Then, we add the entities as vertices to the graph. For each entity added in, only minimal types of the entity are considered adjacent to it. Now the distance between an entity and a type can be defined as the distance of their corresponding vertices in this graph, denoted by $dist(type_i, entity_j)$ (If $type_i$ cannot reach $entity_j$, $dist(type_i, entity_j) = \infty$). The larger distance between a type and an entity is, the less relevant they are. So, we formalize the relevance function as follows:

$$\begin{aligned} f(S_i, x_j) &= rel(type_i, entity_j) \\ &= \Big(\frac{1}{2}\Big)^{dist(type_i, entity_j) - 1}. \end{aligned} \quad (8)$$

As an example, the relevance of an entity with its types is depicted in Fig.2. The membership value of $f(\textit{SovietChessPlayers}, \textit{VasilySmyslov})$ is 1 since they are directed adjacent, and $f(\textit{Player}, \textit{VasilySmyslov}) = \frac{1}{4}$ since the length of shortest directed path between them is 3. $f(\textit{AmericanChessPlayers}, \textit{VasilySmyslov}) = 0$ because that there is no directed path from entity "*VasilySmyslov*" to type "*AmericanChessPlayers*".

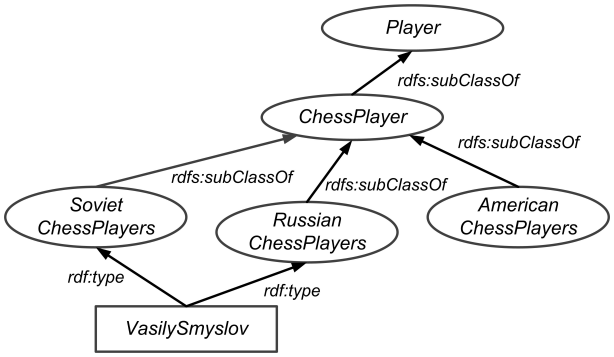

Figure 2: The relevance of an entity with its types

There are many other state-of-the-art approaches that can assess the relevance of an entity with respect to a type (or a facet), such as (Vandic, Frasincar, and Kaymak 2013) and (Liberman and Lempel 2012). We are going to incorporate these approaches in the future.

Evaluation

In this section, we report our evaluation of the proposed approaches, which is conducted on a PC with an Intel Xeon 3.2GHz CPU and 2GB JVM.

Entity Collections

We select the entity collections from Task 1 of the Question Answering over Linked Data campaign (QALD-4).[1] There are 250 questions in total, in which 181 have answer type "Resource". All the resources can be found in DBpdeia 3.9, so do their Yago types as well. Resources without any Yago types or questions having less than 20 answers are removed. The answer lists for the remaining 29 questions are used as entity collections in our evaluation. In average, a collection contains 180 entities (at least 20 entities and at most 1,125 entities), with 8,928 associated types in total. We randomly choose 4 entity collections for parameter tuning, while the rest ones are used for testing.[2]

Task

Given a collection of entities and their types, each participating approach selects no more than K types, which are considered as a group of filters for the given collection. We set $K = 5$ and $K = 8$ in our evaluation. For smaller K, it is not sufficient for filtering, while larger K is difficult for humans to judge its accuracy.

Comparative Approaches

In our evaluation, we compare the BMC-based and EBMC-based approaches, together with 3 alternatives: the CR-based approach, the MMR-based approach, and the one used in Falcons. We briefly introduce them as follows.

[1] http://greententacle.techfak.uni-bielefeld.de/∼cunger/qald/

[2] Detailed information about these collections is available at http://ws.nju.edu.cn/ebmc/

The CR-based approach is widely used in faceted search, which selects types in terms of the descending order of their coverage rates. This approach is considered as the baseline in our evaluation.

The MMR-based approach is an improvement of selecting types by coverage rate. It uses a diversity-based re-ranking technique to increase coverage rate and decrease overlap simultaneously. Let U be the set of all types, S be the set of selected types, and λ be the weighting factor to balance the coverage rate and overlap, MMR can be calculated by the following equation:

$$MMR = \arg\max_{T_i \in U \setminus S} \Big[\lambda \cdot \big(CoverageRate(T_i) - (1-\lambda) \cdot \max_{T_j \in S} Overlap(T_i, T_j)\big)\Big], \quad (9)$$

where $CoverageRate(T_i)$ computes the coverage rate of type T_i, while $Overlap(T_i, T_j)$ computes the Jaccard coefficient of two entity sets covered by types T_i and T_j, respectively. Therefore, MMR maximizes the coverage rate when λ approaches 1, and minimizes the overlap when λ approaches 0. We test $\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$ and set it to 0.3, since it achieves the best performance in our evaluation.

The approach used in Falcons selects types one by one in terms of their coverage rates, and skips a type if it is a subclass of any selected types. If a user had already selected some types as a previous filtering condition, only the subclasses of these types can be further selected. In our evaluation, there is no previous filtering condition so that all types are involved in the candidate list.

The BMC-based and EBMC-based approaches are implemented just as we described in the previous sections. We simply set the ideal coverage rate $R = 1.5/K$, and test parameter α from 1.00 to 3.00, step by 0.25, for various K. Considering the performance of the two approaches on the entity collections, we set $\alpha = 2.00$ for $K = 5$, and $\alpha = 2.50$ for $K = 8$ in this evaluation.

Measurement

Since there are too many types in the candidate type list, it is too hard for human to select K types for each entity collection. Instead, we invite 10 judges who are all CS graduate students to evaluate the quality of the filters generated by different approaches. Each judge is presented with a collection of entities and their types, and is asked to assess filters on 20 entity collections, so that each entity collection with certain K is assessed by 4 judges. They score 1 point to 5 point on the following 5 dimensions:

- Coverage rate ($1 = low$, $5 = high$)
- Overlap ratio ($1 = high$, $5 = low$)
- No. of types ($1 = too\ few$, $5 = almost\ matches\ K$)
- Sensibility ($1 = non\text{-}sensible$, $5 = sensible$)
- Overall ($1 = bad$, $5 = good$)

The dimension "Coverage rate" and "Overlap ratio" are generally considered by almost all the approaches. Since some approaches could select too few types, the dimension "No. of types" is added. The dimension "Sensibility" is particularly designed to assess the difference between the EBMC-based approach and others. A type is sensible if its meaning is accurate, concrete and easy to be understood by humans, which is important to the usefulness of filters. Additionally, we add the "Overall" dimension to achieve comprehensive assessment of the filters.

Table 1: Average scores of the approaches ($K = 5$)

	CR	MMR	Falcons	BMC	EBMC
Coverage rate	4.37 (0.37)	4.57 (0.33)	**4.63** (0.35)	4.37 (0.44)	3.95 (0.48)
Overlap ratio	1.95 (0.42)	2.09 (0.40)	2.68 (0.55)	3.12 (0.50)	**3.39** (0.49)
No. of types	4.18 (0.18)	**4.20** (0.16)	3.79 (0.44)	2.78 (0.57)	3.46 (0.47)
Sensibility	2.75 (0.71)	2.74 (0.60)	2.77 (0.54)	2.63 (0.42)	**3.71** (0.38)
Overall	2.84 (0.45)	2.80 (0.56)	2.96 (0.53)	2.60 (0.40)	**3.55** (0.29)

Table 2: Average scores of the approaches ($K = 8$)

	CR	MMR	Falcons	BMC	EBMC
Coverage rate	4.32 (0.43)	**4.54** (0.41)	4.51 (0.32)	4.33 (0.38)	3.87 (0.40)
Overlap ratio	1.96 (0.49)	2.08 (0.52)	2.80 (0.74)	3.62 (0.61)	**4.00** (0.46)
No. of types	**4.43** (0.18)	**4.43** (0.17)	3.41 (0.56)	3.41 (0.62)	4.11 (0.55)
Sensibility	2.64 (0.66)	2.77 (0.74)	2.73 (0.85)	3.12 (0.58)	**4.14** (0.42)
Overall	2.64 (0.60)	2.73 (0.54)	2.71 (0.67)	3.11 (0.49)	**4.06** (0.41)

Results

Table 1 shows the average scores of the five approaches on different dimensions in the case of $K = 5$, and Table 2 shows the scores for $K = 8$. The average of the standard deviations among different judges for different entity collections is also shown in the table with parentheses. We observe that the results for $K = 5$ and $K = 8$ are quite consistent, and our EBMC-based approach performs best in general among all the five approaches.

By comparing these type selection approaches, we find that the CR-based approach has a high coverage rate, and can provide enough types in most cases. However, it does not consider the problem of overlap, which severely affect the quality of the filters. The MMR-based approach increases the coverage rate and decreases the overlap comparing with the CR-based one, but it still does not work well because it cannot avoid selecting types covering too many entities. The approach in Falcons decreases the overlap to some extent and keeps a relatively high coverage rate, but it fails to select enough types sometimes. Our BMC-based approach decreases the overlap ratio successfully and has a better performance when more types are selected. The EBMC-based approach improves a lot on giving more sensible types with

less overlap, but costs a small loss on coverage rate.

The standard deviations show that the scores given by different judges for the same task are quite similar, which means the judges reach an agreement on each dimension to some extent. Furthermore, Repeated Measures ANOVA indicates that the differences between the EBMC and other approaches in overall are statistically significant ($p < 0.03$) both for $K = 5$ and 8. Additionally, the EBMC-based approach can generate filters per case within 1 second, which demonstrates its efficiency.

We also tried to evaluate different approaches with DBpedia types. Generally speaking, for a large number of entity collections, DBpedia has only less than 10 (even duplicated) types, and most of them cover almost the whole entity collection, which led to similar results between all participating approaches, and it might be unnecessary to select types for filtering in this situation. We are going to consider other type sources for testing our approach in the future.

In overall, the EBMC-based approach is shown as the best to select type filters in our evaluation.

Conclusion

In this paper, we studied how to select types for entity filtering. Our main contributions are as follows:

- We modeled the K type selection problem as the BMC problem, which pursues integral optimization for the coverage quality of type filters. Accordingly, we proposed a cost function on candidate types, which increases the chance of selecting types with appropriate coverage rates.
- We defined a new model, EBMC, which extends the capability of BMC by incorporating fuzzy set. Moreover, we designed an approximation algorithm for EBMC, which achieves the same approximation ratio and time complexity as those of BMC.
- We proposed an EBMC-based approach to selecting type filters, which incorporates the relevance between entities and types by leveraging type hierarchy. Our evaluation results show that the EBMC-based approach tends to select sensible types with superior quality.

In future work, we look forward to applying EBMC to other problems such as entity summarization. We also want to design new algorithms for EBMC with a higher approximation ratio while keeping an acceptable time complexity.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61170068, 61223003 and 61370019. We would like to thank Dr. Gong Cheng for his valuable suggestion on the design of our experiments, and Yong Wang for his help on evaluation. We are also grateful to all participants in the evaluation for their time and effort.

Appendix A: Proof of Approximation Ratio and Running Time of Algorithm 1

Proof of Lemma 1.

$$\begin{aligned} w(A \cup B) &= \sum_{j=1}^{n} \big(w_j \cdot \max\{f(P, x_j) \mid P \in A \cup B\}\big) \\ &\le \sum_{j=1}^{n} \big(w_j \cdot \max\{f(P, x_j) \mid P \in A\}\big) \\ &\quad + \sum_{j=1}^{n} \big(w_j \cdot \max\{f(P, x_j) \mid P \in B\}\big) \\ &= w(A) + w(B). \end{aligned}$$

□

Proof of Lemma 2.

$$\begin{aligned} & w(A) - w(B) \\ =& \sum_{j=1}^{n} \Big(w_j \cdot \big(\max_{Q \in A} f(Q, x_j) - \max_{Q \in B} f(Q, x_j)\big)\Big) \\ \le& \sum_{j=1}^{n} \Big(w_j \cdot \sum_{P \in A \setminus B} \big(\max_{Q \in B \cup \{P\}} f(Q, x_j) - \max_{Q \in B} f(Q, x_j)\big)\Big) \\ =& \sum_{P \in A \setminus B} \big(w(B \cup \{P\}) - w(B)\big). \end{aligned}$$

□

Proof of Lemma 3.
According to Lemma 2, we have $w(OPT) - w(G_{i-1}) \le \sum_{P \in OPT \setminus G_{i-1}} \big(w(G_{i-1} \cup \{P\}) - w(G_{i-1})\big)$. Because S_{G_i} is selected by the greedy strategy, we have $\frac{w(G_{i-1} \cup \{P\}) - w(G_{i-1})}{c(\{P\})} \le \frac{w(G_i) - w(G_{i-1})}{c(\{S_{G_i}\})}$ for each $P \in OPT \setminus G_{i-1}$. Thus,

$$\begin{aligned} & w(G_i) - w(G_{i-1}) \\ =& \frac{c(\{S_{G_i}\})}{c(OPT)} \cdot \frac{\sum_{P \in OPT} c(\{P\})}{c(\{S_{G_i}\})} \cdot \big(w(G_i) - w(G_{i-1})\big) \\ \ge& \frac{c(\{S_{G_i}\})}{L} \cdot \sum_{P \in OPT \setminus G_{i-1}} \Big(\frac{c(\{P\})}{c(\{S_{G_i}\})} \cdot \big(w(G_i) - w(G_{i-1})\big)\Big) \\ \ge& \frac{c(\{S_{G_i}\})}{L} \cdot \sum_{P \in OPT \setminus G_{i-1}} \big(w(G_{i-1} \cup \{P\}) - w(G_{i-1})\big) \\ \ge& \frac{c(\{S_{G_i}\})}{L} \cdot \big(w(OPT) - w(G_{i-1})\big). \end{aligned}$$

□

Proof of Lemma 4.
Basis. When $i = 1$, $w(\{S_{G_1}\}) \ge \frac{c(\{S_{G_1}\})}{L} \cdot w(OPT)$, because the ratio $\frac{w(\{S_{G_1}\})}{c(G_1)}$ is maximum over all sets.

Inductive hypothesis. For $i = 1, \ldots, p-1$, we have $w(G_i) \ge \Big(1 - \prod_{k=1}^{i} \big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT)$.

Inductive step. When $i = p$,

$$
\begin{aligned}
&w(G_p) = w(G_{p-1}) + \big(w(G_p) - w(G_{p-1})\big) \\
\geq &w(G_{p-1}) + \frac{c(\{S_{G_p}\})}{L} \cdot \big(w(OPT) - w(G_{p-1})\big) \\
= &\big(1 - \frac{c(\{S_{G_p}\})}{L}\big) \cdot w(G_{p-1}) + \frac{c(\{S_{G_p}\})}{L} \cdot w(OPT) \\
\geq &\big(1 - \frac{c(\{S_{G_p}\})}{L}\big)\Big(1 - \prod_{k=1}^{p-1}\big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT) \\
&+ \frac{c(\{S_{G_p}\})}{L} \cdot w(OPT) \\
= &\Big(1 - \prod_{k=1}^{p}\big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT).
\end{aligned}
$$

Thus, $w(G_i) \geq \Big(1 - \prod_{k=1}^{i}\big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT)$. □

Proof of Theorem 1.
Case 1. For some $S_k \in S$, $w(\{S_k\}) \geq \frac{1}{2} \cdot w(OPT)$. We have $\frac{w(\{T\})}{w(OPT)} \geq \frac{w(\{S_k\})}{w(OPT)} \geq \frac{1}{2}$.

Case 2. For all $S_k \in S$, $w(\{S_k\}) < \frac{1}{2} \cdot w(OPT)$.

Case 2.1 $c(G) < \frac{L}{2}$. Then $\forall S_k \notin G$. $c_k > \frac{L}{2}$, implying that there is at most one fuzzy set in $OPT \setminus G$ because of $c(OPT) \leq L$. We have $w(G) \geq w(OPT \cap G) \geq w(OPT) - w(OPT \setminus G) > \frac{1}{2} \cdot w(OPT)$.

Case 2.2 $c(G) \geq \frac{L}{2}$. According to Lemma 4, we have $w(G_i) \geq \Big(1 - \prod_{k=1}^{i}\big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT)$. Because $c(G) = \sum_{i=1}^{l} c(\{S_{G_i}\}) \geq \frac{L}{2}$. We have $\Big(1 - \prod_{k=1}^{l}\big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \geq \Big(1 - \big(1 - \frac{1}{2l}\big)^l\Big)$, which implies

$$
\begin{aligned}
w(G_l) &\geq \Big(1 - \prod_{k=1}^{l}\big(1 - \frac{c(\{S_{G_k}\})}{L}\big)\Big) \cdot w(OPT) \\
&\geq \Big(1 - \big(1 - \frac{1}{2l}\big)^l\Big) \cdot w(OPT) \\
&\geq \big(1 - \frac{1}{\sqrt{e}}\big) \cdot w(OPT).
\end{aligned}
$$

So, the weight of the collection generated by Algorithm 1 is proved to be at least $\big(1 - \frac{1}{\sqrt{e}}\big) \cdot w(OPT)$ in each of the cases. □

Proof of Theorem 2.
In each iteration, we need to check all the fuzzy sets in U to get S_{G_i}, which means no more than m fuzzy sets will be checked. Regarding the time complexity of adding a new fuzzy set P into S', the addition weight $w(S' \cup \{P\}) - w(S')$ can be calculated by $\sum_{j=1}^{n}\big(w_j \cdot \max\big\{f(Q, x_j) \mid Q \in (S' \cup \{P\})\big\}\big) - \sum_{j=1}^{n}\big(w_j \cdot \max\big\{f(Q, x_j) \mid Q \in S'\big\}\big)$, which can be done in $O(n)$ time. Thus, each iteration takes $O(mn)$ time, and the running time in total is $O(m^2 n)$ since U will be empty after m iterations. □

References

Carbonell, J., and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 335–336. ACM Press.

Cheng, G., and Qu, Y. 2009. Searching linked objects with Falcons: Approach, implementation and evaluation. *International Journal on Semantic Web and Information Systems* 5(3):49–70.

Cohen, R., and Katzir, L. 2008. The generalized maximum coverage problem. *Information Processing Letters* 108(1):15–22.

Hearst, M. A. 2006. Clustering versus faceted categories for information exploration. *Communications of the ACM* 49(4):59–61.

Khuller, S.; Moss, A.; and Naor, J. 1999. The budgeted maximum coverage problem. *Information Processing Letters* 70(1):39–45.

Liberman, S., and Lempel, R. 2012. Approximately optimal facet selection. In *Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, March 26-30, 2012*, 702–708.

Nie, Z.; Ma, Y.; Shi, S.; Wen, J.; and Ma, W. 2007. Web object retrieval. In *Proceedings of 16th International World Wide Web Conference*, 81–90. ACM Press.

Sah, M., and Wade, V. 2013. Personalized concept-based search and exploration on the web of data using results categorization. In *Proceedings of 10th Extended Semantic Web Conference*, 532–547. Springer.

Tonon, A.; Catasta, M.; Demartini, G.; Cudre-Mauroux, P.; and Aberer, K. 2013. TRank: Ranking entity types using the web of data. In *Proceedings of 12th International Semantic Web Conference*, 640–656. Springer.

Vandic, D.; Frasincar, F.; and Kaymak, U. 2013. Facet selection algorithms for web product search. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, 2327–2332.

Xu, Z.; Fu, Y.; Mao, J.; and Su, D. 2006. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of WWW 2006 Workshop on Collaborative Web Tagging*.