Proceedings of the Federated Conference on Computer Science and Information Systems pp. 115–122
DOI: 10.15439/2019F2
ISSN 2300-5963 ACSIS, Vol. 18

A Minimum Set-Cover Problem with several constraints

Jens Dörpinghaus*, Carsten Düing†, Vera Weil‡
Fraunhofer Institute for Algorithms and Scientific Computing,
Schloss Birlinghoven, Sankt Augustin, Germany
Email: *jens.doerpinghaus@scai.fraunhofer.de, †carsten.cdueing@scai.fraunhofer.de
‡Department for Computer Science,
University of Cologne, Germany
Email: weil@informatik.uni-koeln.de

***Abstract*—A lot of problems in natural language processing can be interpreted using structures from discrete mathematics. In this paper we will discuss the search query and topic finding problem using a generic context-based approach. This problem can be described as a Minimum Set Cover Problem with several constraints. The goal is to find a minimum covering of documents with the given context for a fixed weight function. The aim of this problem reformulation is a deeper understanding of both the hierarchical problem using union and cut as well as the non-hierarchical problem using the union. We thus choose a modeling using bipartite graphs and suggest a novel reformulation using an integer linear program as well as novel graph-theoretic approaches.**

I. Introduction

In scientific research, expert systems provide users with several methods for knowledge discovery. They are widely used to find relevant or novel information. For example, medical and biological researchers try to find molecular pathways, mechanisms within living organisms or special occurrences of drugs or diseases. In [1], we discussed a novel approach for describing NLP problems using theoretical computer science. Using this approach, it is possible to obtain the algorithmic core of a NLP problem. Here, we will discuss two $\mathcal{NP}$-complete problems: Search Query Finding (SQF) and Topic Finding (TF).

Using expert system as an input, researches usually consider an initial idea and some content like papers or other documents. The most common approach is inquiring a search engine to find closely related information. Thus two question are most frequently asked: "How can I find these documents?" to adjust the search query for knowledge discovery or "What are these documents all about?" to find the topic. Both questions are heavily related to the context of documents. Meta-data like authors, keywords and text are used to retrieve results of a query using a search engine.

Semantic searches are usually based on textual data and some meta-data like authors, journals, keywords. In addition, time and complexity play an important role, since often relevant information is not findable or new information is already available. For example, databases such as PubMed [2] contain around 27 million abstracts and PMC[1] includes around 2 million biomedical-related full-text articles.

Both problems are equivalent (see [1]) and can be described as a Minimum Set Cover Problem with several constraints. Query languages and natural languages are not only highly connected but merge more and more (see [3] or [4]). The goal is to find a minimum covering of documents with the given context for a fixed weight function. The aim of this problem reformulation is a deeper understanding of both the hierarchical as well as the non-hierarchical problem. We thus choose a modeling using bipartite graphs and suggest a novel reformulation using an integer linear program as well as graph-theoretic approaches.

There is a considerable amount of literature on both problems. Many studies have been published on probabilistic or machine-learning-approaches, see [5], [6] or [7]. In addition, in recent years there has been growing interest in providing users with suggestions for more specific or related search queries, see [8].

This paper is divided into six sections. The first section gives a brief overview of the problem formulation and provides the definition of MDC and WMDC. The second section analyses the hierarchical problem formulation and proposes novel heuristics. In the third section, we present a short analysis of the non-hierarchical problem and propose an integer linear program approach and some modified graph heuristics to solve this problem. We present some experimental results on artificial and real-world scenarios in section four. Our conclusions are drawn in the final section.

II. Problem Formulation And Definition

We follow the notation introduced in [1]. Let $\mathbb{D}$ be a set of documents and let $\mathbb{X}$ be a set of context data. Context data is information associated with documents, such as keywords, authors, publication venue, etc. Both $\mathbb{D}$ and $\mathbb{X}$ form the vertex set of a graph G. If and only if a description of a document $d \in \mathbb{D}$ is associated with context data $x \in \mathbb{X}$, we add the edge $\{d, x\}$ to E. The graph $G = (\mathbb{D} \cup \mathbb{X}, E)$ is bipartite and called *document description graph*.

[1] https://www.ncbi.nlm.nih.gov/pmc/

Given a subset $R \subset \mathbb{D}$, the search-query-finding (SQF) or topic-finding (TF) problem tries to find a good description of R with terms in $\mathbb{X}$. In general, we lack a proper definition of what *good* means.

For example, given a search engine $q : \mathbb{X} \to \mathbb{D}$ and a description function $f : \mathbb{D} \to \mathbb{X}$, we want a solution $Z \subset \mathbb{X}$ such that $q(Z) = R$ and $Z = f(R)$. If we want to obtain a human-readable topic for R, we need a solution Z of minimum cardinality which precisely describes all documents in R, hence distinguishing R from $\mathbb{D} \setminus R$ without duplication and redundancies. See Figure 1 for an illustration of the relation between the sets X, R and the mappings f, q.

To sum up, we need to find a minimum covering of R with elements in $\mathbb{X}$ so that whenever we are forced to cover further documents, that is, documents in $\mathbb{D} \setminus R$, the number of these further documents is minimal. Depending on the considered problem and the usecase, we have to make a trade-off between the size of the subset in $\mathbb{X}$ and the number of covered documents in $\mathbb{D} \setminus R$. However, these problems are all related to the problem of finding dominating sets in bipartite graphs, see [9]. The latter is $\mathcal{NP}$-complete, even for bipartite graphs, see [10].

For $x_i \in \mathbb{X}$, we call $D_i = N(x_i) \subseteq \mathbb{D}$ the cover set of x_i in $\mathbb{D}$. Roughly speaking, just imagine a keyword x_i and all associated documents D_i. With this, we reformulate the problem as follows:

Definition II.1. *(Document Cover Problem, DC) Let $\mathbb{D}$ be a set of documents, let $\mathbb{X}$ be a set of context data and let $G = (\mathbb{D} \cup \mathbb{X}, E)$ be the document description graph.*

Given a set of documents $R \subset \mathbb{D}$, a solution of the DC is a set $C \subseteq D$ that covers at least R.

Definition II.2. *(Minimum Document Cover Problem, MDC) Let C be a solution of the DC and let $\alpha_2 = |C|$. Let further $\alpha_1 = r$ be the number of documents in $C \setminus R$.*
A solution of MDC is a solution of DC so that $\alpha = \alpha_1 + \alpha_2$ is minimal.

We can define two objectives for minimization: α_1 and α_2.

Definition II.3. *(α_2-Minimum Document Cover Problem, α_2-MDC) Given a set of documents $R \subset \mathbb{D}$, a solution of the α_2-MDC is a solution of DC so that $\alpha = \alpha_1$ is minimal.*

Definition II.4. *(α_1-Minimum Document Cover Problem, α_1-MDC) Given a set of documents $R \subset \mathbb{D}$, a solution of the α_1-MDC is a solution C of DC so that $\alpha = \alpha_2$ is minimal.*

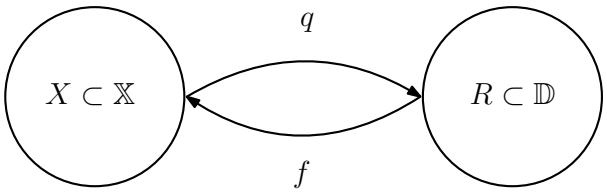

Fig. 1: Relation between the sets $X \subset \mathbb{X}$ as description set of documents in $R \subset \mathbb{D}$.

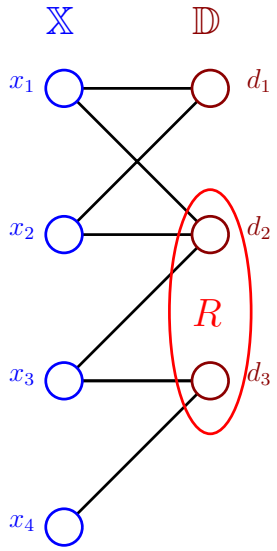

Fig. 2: A graph $G = (\mathbb{D} \cup \mathbb{X}, E)$ illustrating example II.8

We further introduce a weighted version of this problem:

Definition II.5. *(Weighted Minimum Document Cover Problem, WMDC) Let $\mathbb{D}$ be a set of documents, let $\mathbb{X}$ be a set of context data and let $G = (\mathbb{D} \cup \mathbb{X}, E)$ the document description graph. Let $w : \mathbb{X} \to \mathbb{R}$ be a weight function which associates a weight for every element in $\mathbb{X}$. Moreover, we set $D = \{D_1, ..., D_n\}$. Let $\alpha_1 = r$ be the number of documents in $R \subset \mathbb{D}$ and $\alpha_2 = |C|$.*
A solution of the WMDC is a set $C \subseteq D$ which covers R, such that $\alpha = \alpha_1 + \alpha_2 + w(C)$ is minimal, where $w(C) = \sum_{c \in C} w(c)$.

Again we can find formulations for α_1-WMDC and α_2-WMDC. Both problems are $\mathcal{NP}$-hard, see [11].

In general, we will focus on the α_2 optimization. Thus, in this paper, we denote this version with the MDC and WMDC.

We have to distinguish between hierarchical and non-hierarchical approaches. Both MDC and WMDC search for a cover set $c_1, ..., c_n$ which leads to a solution $c_1 \cup ... \cup c_n$. This is a non-hierarchical approach. Using a search engine this would lead to a solution c_1 `or ... or` c_n. Utilizing the cut of sets we will need a hierarchical solution $(c_1 \cup ... c_n) \cap (c_{n+1} \cup ... c_m) \cap ...$. Using a search engine would lead to a solution (c_1 `or ... or` c_n) `and` (c_{n+1} `or ... or` c_m)) `and` ....

Definition II.6. *(Hierarchical Minimum Document-Cover Problem, HMDC) Let $\mathbb{D}$ be a set of documents, let $\mathbb{X}$ be a set of context data and let $G = (\mathbb{D} \cup \mathbb{X}, E)$ be the document description graph. Moreover, we set $D = \{D_1, ..., D_n\}$.*
A solution of the HMDC problem for $R \subset \mathbb{D}$ is a minimum cover $C \subseteq D$ with $C = C_1 \cap ... \cap C_n$ and $C_i = C_1^i \cup ... \cup C_m^i$ of R so that $C \setminus R$ is minimal. We use $N(x_i)$ as usual for the open neighborhood $N(x_i) \setminus x_i$.

Definition II.7. *(Hierarchical Weighted Minimum Document-*

Cover Problem, HWMDC) Given a set of documents $\mathbb{D}$, a set of context data $\mathbb{X}$ and the document description graph $G = (\mathbb{D} \cup \mathbb{X}, E)$. We set $D = \{D_1, ..., D_n\}$. Given a weight function $w : \mathbb{X} \to \mathbb{R}$ that defines a weight for every element in $\mathbb{X}$.

A solution of the weighted HWMDC problem for $R \subset \mathbb{D}$ is a minimum cover $C \subseteq D$ with $C = C_1 \cap ... \cap C_n$ and $C_i = C_1^i \cup ... \cup C_m^i$ of R, i.e. $\sum_{c \in C} w(c)$ is minimal, so that $C \setminus R$ is minimal.

We will discuss two examples for the non-hierarchical problem:

Example II.8. *Given an instance of the MDC with $\mathbb{D} = \{d_1, d_2, d_3\}$, $R = \{d_2, d_3\}$, $\mathbb{X} = \{x_1, ..., x_4\}$ and $D_1 = D_2 = \{d_1, d_2\}$, $D_3 = \{d_2, d_3\}$, $D_4 = \{d_3\}$. See figure 2 for an illustration.*

A minimum set cover cannot include x_1 or x_2, but a solution is $C = D_3$.

Example II.9. *Consider the instance given in example II.8 with additional weights $w(x_1) = w(x_2) = w(x_3) = 1$ and $w(x_4) = 0$. A minimum solution of the weighted MDC can be found with $Z = \{x_3, x_4\}$.*

Let $w(x_1) = w(x_3) = 1$ and $w(x_4) = w(x_2) = 0$. A minimum solution of weighted MDC can be either found with $Z = \{x_2, x_4\}$, here $w(Z) = 0$ but $|C \setminus R| = 1$. If we chose $Z = \{x_3, x_4\}$ $w(Z) = 1$ but $|C \setminus R| = 0$.

We will first of all focus on hierarchical approaches, discussing appraoches using dynamic programming and bipartite graph heuristics or spanning trees. After that we will discuss the non-hierarchical problem and solutions using an integer linear program approach as well as some heuristics utilizing the graph structure. We will evaluate the results on some random instances and finish with a conclusion.

III. Hierarchical approaches

A. Problem Description

For some questions it is interesting to find a cover of $R \subset \mathbb{D}$ with increasing (decreasing) or selectable exactness and the number of named entities $Z \subset X = f(R)$. If we have a set of documents and want to obtain more others closely related documents, we may be interested in a modification of the similarity measure for documents or search queries. We build covers $C_i = q(Z_i)$ of R and optimize the solution by concatenating them with a logical AND.

B. Using unique keyword descriptions on bipartite graphs

From the graph in figure 2 we can see that the graph $G = (\mathbb{D} \cup \mathbb{X}, E)$ is bipartite. The neighborhood $N(d) \subset \mathbb{X}$ of every document $d \in \mathbb{D}$ is not necessarily unique description of this document. Thus we can find a trivial solution of the MDCP on $R \subset \mathbb{D}$ by

$$\vee_{d \in R}(\wedge_{x \in N(d)} x)$$

We can eliminate elements with the largest error from this list. This process can be limited by iterations as well as a precision. For example we may limit the precision to 0.9 which will eliminate at maximum 10% of all keywords, whereas a precision of 0.5 will eliminate at maximum 50%.

Algorithm 1 KEYWORD-COVER

Require: Documents $\{d_1, ..., d_n\} \subset \mathbb{D}$ and descriptive elements $f(d_i) = \{x_1, ..., x_m\} \subset \mathbb{X}$, a weight function $w : \mathbb{X} \to \mathbb{R}$ maxiter as maximum of iterations, prec as precision

Ensure: A cover $Z = (x_i \wedge x_j \wedge ...) \vee (x_k \wedge x_l \wedge ...) \vee ...$ of R with elements in $\mathbb{X}$.

$f' = f$
2: **for** every $d \in R$ **do**
  while iteration<maxiter AND $f'(d) > (\text{prec} \cdot f(d))$ **do**
4: remove $x \in f'(d)$ with maximum weight
  end while
6: **end for**
  return $Z = \vee_{d \in R}(\wedge_{x \in f'(d)} x)$

If we set $w : \mathbb{X} \to \mathbb{R}$ as the error function $err(x) = |q(x) \setminus R|$ we will find a solution for MDCP, otherwise this will return a solution of WMDCP. The function err is a less time-consuming approach but highly depended on the distribution of $\mathbb{X}$.

C. Dynamic programming and bipartite graph heuristic

Here, we describe a heuristic and dynamic method by creating dominating subgraphs of a bipartite graph. Building the bipartite graph $G_b = (V = R \cup X, E)$, a subgraph of the document description graph $G = (\mathbb{D} \cup \mathbb{X}, E)$, we create a set with documents $R_a = \{d_1, ..., d_n\} \subseteq \mathbb{D}$ and all their context data (like keywords, named entities, etc.) in a sorted list $X_a = \{x_1, ..., x_m\} \subseteq \mathbb{X}$ for the two sets of nodes. The edges (d_i, x_j) in G_b are given for all pairs d_i, x_j iff $x_j \in f(d_i)$. The elements in X_a should be sorted ascending or descending by their degree. For our example we choose a descending order, which results in an increasing precise cover.

In addition we need to build a second set R_b as temporary storage for the documents and a sorted list of lists $Z = \{Z_1, Z_2, \ldots, Z_k\}$, with the covers Z_i of R_a for the output. The algorithm in pseudocode can be found in alg. 2. In every execution of the while loop in line 7 a new sublist $Z_i \subset Z$ is created (see line 13). All of them are complete covers of all documents in R_a, where Z_0 may contain just one element x_i with $N(x_i) = R_a$ and the last Z_m may contain just all identities, that means x_i with a single neighbor $N(x_i) = d_i$. There are many options to modify the algorithm for special use cases. Choosing the ascending order for X_a and the minimum in line 9, which is same as in the other case just means the first $x_j \in X_a$, will mostly give different results.
If after the last run of the loop X_a is empty, but there are still documents in R_a, we receive an incomplete cover Z_k. To avoid that we add the ID's for the last documents in R_a (in descendind order) to Z_k, or create and add an all covering x_∞ (for descending order).

Algorithm 2 HIERARCHICAL BIPARTITE COVER-DESCRIPTION

Require: Documents $\{d_1, ..., d_n\} \subset \mathbb{D}$ and descriptive elements $f(d_i) = \{x_1, ..., x_m\} \subset \mathbb{X}$, R_a with all d_i and empty set R_b, sorted list X with all x_i and empty list Z, $G = (R_a \cup X_a, E)$ with $(d_i, x_j) \in E$ if $d_i \in l(x_j)$, order: descending or ascending, maximum iterations maxdeep

Ensure: List of covers Z of $R_a = \{d_1, ..., d_n\}$ with elements in $\mathbb{X}$.

for every $x_i, x_j \in X$ **do**
  if $N(x_i) = N(x_j)$ **then**
    $x_i = \{x_i$ OR $x_j\}$, remove x_j
  end if
end for
k←0
while $|X| > 0$ AND $k \leq$maxdeep **do**
  for every $d \in R_a$ **do**
    choose $x_j \in N(d_i)$ with $max|N(x_j)|$ (or min at ascending)
    for every $d \in N(x_j)$ **do**
      $R_b \leftarrow d$, from R_a.remove(d)
    end for
    move x_j to Z_k
  end for
  $R_a = R_b$, $R_b = \emptyset$, $k = k + 1$
end while
if $R_a \neq \emptyset$ **then**
  if (order = ascending): add x_∞ to last Z_k
  if (order = descending): add $f(d_i)$ for all $d_i \in R_a$ to last Z_k
end if
return $Z = \{Z_1$ AND ... AND $Z_k\}$

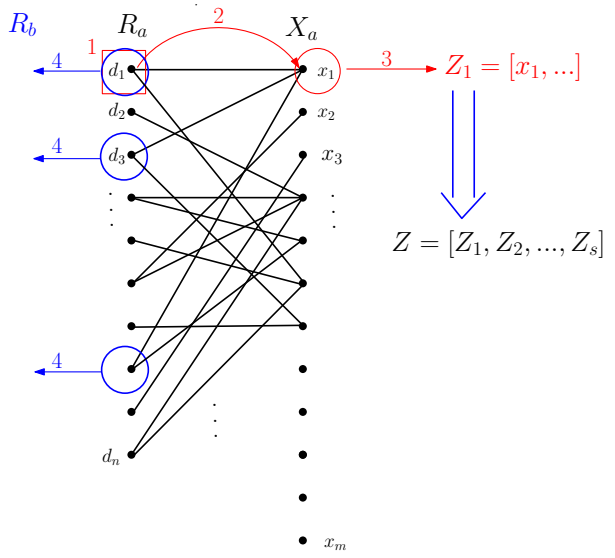

Fig. 3: Illustration of the bipartite graph algorithm.

D. Spanning Tree Appraoch

Given a set of documents $\mathbb{D}$, a set of context data $\mathbb{X}$ and the document description graph $G = (\mathbb{D} \cup \mathbb{X}, E)$. We can define

Algorithm 3 TREE-DESCRIPTION

Require: Documents $d_1, ..., d_n \subset \mathbb{D}$ and descriptive elements $f(d_i) = \{x_1, ..., x_m\} \subset \mathbb{X}$

Ensure: A spanning tree S describing $R = \{d_1, ..., d_n\}$ with elements in $\mathbb{X}$.

build list $x_i : l(x_i)$ with $i \in \{1, ..., m\}$ and $l(x_i) = q(x_i)$
build $G = (X, E)$ with $X = \{x_i, ..., x_m\}$ and $(x_i, x_j) \in E$ iff $l(x_j) \subset l(x_i)$ and weight $w(x_i, x_j) = |l(x_i)| - |l(x_j)|$
$m = \max_{x \in X} l(x)$
$X = X \cup x_o$
for every $x \in X$ with $l(x) = m$ **do**
  add edge (x_0, x)
end for
Calculate Minimum Spanning Tree S in G
return S

$\forall x_i \in \mathbb{X}$ $D_i = N(x_i)$ as the cover set of x_i in $\mathbb{D}$. We set $D = \{D_1, ..., D_n\}$.

A solution of the MDC problem for $R \subset \mathbb{D}$ is a minimum cover $C \subseteq D$ of R so that $C \setminus R$ is minimal.

We can now construct a hierarchical tree using the logical operators *and* and *or* in $\mathbb{X}$. We will do this by considering a directed graph $G' = (V, E)$ with nodes $V = \mathbb{X}$. We add weighted edges between two nodes x_i, x_j if $N_G(x_j) \subset N_G(x_i)$. The weight is set to $w(x_i, x_j) = |N_G(x_i)| - |N_G(x_j)|$. If we add a meta node x_0 that is connected to all nodes that have no nodes adjacent to them, which means to all nodes x with $\delta_G^-(x) = 0$, we can search for minimum spanning trees, see figure 4.

Finding the spanning tree(s) in this graph G' can be done using breadth-first search (BFS) or depth-first search (DFS) in $O(|V| + |E|)$ time. Finding the minimum spanning tree can also be done using this approach since the edges are sorted according to their weight. This a a technical assumption and we will have different findings on different definitions of $\mathbb{X}$. Finding minimum spanning trees is in general $\mathcal{NP}$-complete, see [12]. See algorithm 3 for pseudocode.

As we can see, even this simple approach needs a complex heuristic. Although finding minimum spanning trees is usually in $\mathcal{FP}$, we can construct more complex examples that are $\mathcal{NP}$-complete. It would be very beneficial to find problems that are in $\mathcal{P}$.

IV. NON-HIERARCHICAL APPROACHES

A. Problem Description

Looking for non-hierarchical approaches we want to find a minimum cover $C \subset D$ wihtout step by step optimization by connecting partial results with logical AND. We here present two ways to do this, first by using an integer linear program and second by using a small modification of the bipartite graph algortihm.

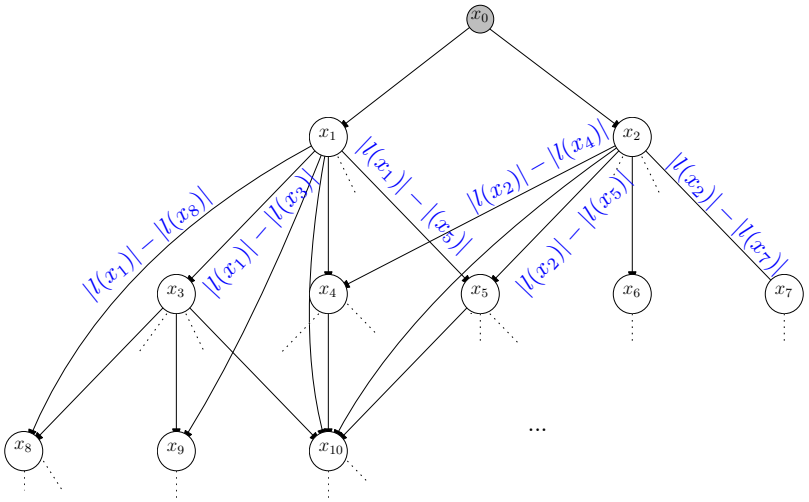

Fig. 4: Illustration of set representative in the graph $G' = (V, E)$ and weight $w(x_i, x_j)$ after adding the meta node x_0, with $l(x_i) := |N_G(x_i)|$. Not all edges and nodes have been added.

B. An Integer Linear Program Approach

Numerous ILP-formulations for the set-cover problem can be found in literature, for example [13] or [14]. To meet definition II.6 of MDC we need to adjust the formulation.

Given a set of documents $\mathbb{D}$, a subset $R \subset \mathbb{D}$, a set of context data $f(R) = X \subset \mathbb{X}$ and the document description graph $G = (\mathbb{D} \cup \mathbb{X}, E)$. We can define $\forall x_i \in \mathbb{X}\ D_i = N(x_i)$ as the cover set of x_i in $\mathbb{D}$. We set $D = \{D_1, ..., D_n\}$ and $e(D_i) = D_i \setminus R$ as the error of the description term x_i.

A solution of the MDC problem for $R \subset \mathbb{D}$ is a minimum cover $C \subseteq D$ of R so that $C \setminus R$ is minimal.

$$\begin{array}{lrcl} \min & \sum_{i=1}^{n} x_i & + & \sum_{i=1}^{n} x_i e(X_i) \\ \text{subject to} & \sum_{i:v\in X_i} x_i & \geq & 1,\ \forall v \in R \\ & x_i & \geq & 1\ \ \forall i = 1, ..., n \\ & x_i & \in & \mathbb{Z}\ \ \forall i = 1, ..., n \end{array} \tag{1}$$

Here the vector x gives a set $Z \subset X$ which gives a minimum cover $q(Z) = C \subset D$ of R so that $C \setminus R$ is minimal.

The weighted MDC problem was introduced in definition II.7. Given a weight function $w : \mathbb{X} \to \mathbb{R}$ that defines a weight for every element in $\mathbb{X}$ the ILP (1) changes as follows:

$$\begin{array}{lrcl} \min & \sum_{i=1}^{n} w(x_i) & + & \sum_{i=1}^{n} x_i e(X_i) \\ \text{subject to} & \sum_{i:v\in X_i} x_i & \geq & 1\ \ \forall v \in R \\ & x_i & \geq & 1\ \ \forall i = 1, ..., n \\ & x_i & \in & \mathbb{Z}\ \ \forall i = 1, ..., n \end{array} \tag{2}$$

A solution of the MDC problem for $R \subset \mathbb{D}$ is a minimum cover $C \subseteq D$ of R, i.e. $\sum_{c\in C} w(c)$ is minimal, so that $C \setminus D$ is minimal.

C. Dynamic programming and bipartite graph heuristic

We can use algorithm 2 to construct a non-hierarchical solution. This algorithm has already been used to computed k covers of R_a, which can be used to find a cover with minimal error $Z = \min_{e(x_i)} Z_i$, that means for $q(Z) = C\ C \setminus R$ is minimal. The pre-sorting of the context data list X results in covers of ascending cardinality, so the number of iterations k may be a limit for maximum cardinality. The pre-sorting can be removed, which results in more balanced and random

Algorithm 4 BIPARTITE COVER-DESCRIPTION

Require: Documents $\{d_1, ..., d_n\} \subset \mathbb{D}$ and descriptive elements $f(d_i) = \{x_1, ..., x_m\} \subset \mathbb{X}$, R_a with all d_i and empty set R_b, sorted list X with all x_i and empty list C, $G = (R_a \cup X_a, E)$ with $(d_i, x_j) \in E$ if $d_i \in N(x_j)$, maximum iterations maxdeep

Ensure: A minimum covers Z of $R_a = \{d_1, ..., d_n\}$ with elements in $\mathbb{X}$.

for every $x_i, x_j \in X$ **do**
  if $N(x_i) = N(x_j)$ **then**
    $x_i = \{x_i$ OR $x_j\}$, remove x_j
  end if
end for
k←0
while $|X| > 0$ AND $k \leq$maxdeep **do**
  for every $d \in R_a$ **do**
    choose $x_j \in N(d_i)$ with $max|N(x_j)|$
    for every $d \in N(x_j)$ **do**
      $R_b \leftarrow d$, from R_a.remove(d)
    end for
    move x_j to Z_k
  end for
  $R_a = R_b$, $R_b = \emptyset$, $k = k + 1$
end while
if $R_a \neq \emptyset$ **then**
  add x_∞ to last Z_k
end if
return $Z = \min_{i\in\{1,...,k\}} Z_i$,

covers, whereof one with minimum error can be chosen.

V. Experimental Results

We tested our novel approach within two scenarios. First of all, using an artificial random instances with $|\mathbb{D}| = 150$ documents and a given subset R with 20 example documents. We created instances with a fixed number of 80 or 40 normal distributed keywords which had a significant impact on the output. In addition we used N iterations, which lead to a different precision. The second scenario is a real-world example using set R of 10 random documents out of a human curated topic. We tested against complete PubMed Database using SCAIView. Thus $|\mathbb{D}| \approx 29,000,000$.

Within the random instances we were unable to describe a single document by its random keywords. This approach usually returned more than 100 documents. The reason for this rather contradictory result is still not entirely clear, but the normal distribution of keywords may be responsible for this result. The algorithms Tree-Description and Hierarchical Bipartite Cover-Description performed quite well, see figure 5. In general, we found Hierarchical Bipartite Cover-Description to work better and faster.

Changing to the real-world scenario the situation changes significantly. Given a set of 10 documents, Hierarchical Bipartite Cover-Description usually returned more than 6,000,000 documents, Tree-Description more than 5,000,000 before reaching the search-query length limitations. Vice versa we found, that the combination of keywords described a single document very well – even within nearly 3 million documents in $\mathbb{D}$. The keywords using MeSH-terms in PubMed are manually curated and seem not to be normally distributed.

The output of Keyword-Cover for 10 random examples with $|R| = 10$ is presented in figures 6 and 7. The precision was iterated from 0.9 to 0.4. The output scales very well and is quite stable till precision 0.5 where we found between 12 and 36 documents. For precision 0.4 we found 28 till 676 documents.

We can see, that we have found a novel solution for search query finding on literature that performs quite well on real-world data. Our work clearly has some limitations. It is not clear, why the proposed algorithms perform significantly different in both scenarios. Despite this we believe our work could be the basis for solving the SQF and TD. Further work needs to be performed to the distribution of descriptive elements to documents to establish whether they can be used to generate search queries and topic descriptions that are significant enough.

VI. Conclusions

We presented a novel formulation of both search query and topic finding problems as Minimum Set-Cover Problems. We proposed a weighted and unweighted version of the Minimum Document-Cover Problem as well as a hierarchical version using both `AND` as well as `OR` and the non-hierarchical version only using and.

With this we get a solution that uses on the one hand as much descriptive elements as possible to get as less documents in $\mathbb{D}$ but not in R.

The search queries are not human readable. For example the tree-approach returns queries in the form `MeSH_Terms: D000818" AND ("MeSH_Terms: D051381" OR "MeSH_Terms: D009538" OR "MeSH_Terms: D017207" OR "MeSH_Terms: Q000494" OR "MeSH_Terms: D006624" OR "MeSH_Terms: D011978" OR "MeSH_Terms: D000109" OR "MeSH_Terms: D008297" OR "MeSH_Terms: Q000187" OR "MeSH_Terms: Q000502" OR "MeSH_Terms: Q000378" OR "MeSH_Terms: D008464" OR "MeSH_Terms: Q000187" OR "MeSH_Terms: Q000187" OR ....` This can be easily translated into something human-readable. But still it is a good probability that further research has to be done on how to shorten this to be both precise as well as significant.

In general this is both: a correct solution of clustering labeling of R on $\mathbb{X}$ obtained by f as well as a possible solution of a search query so that $q(Z) = R$. It is not necessary an optimal solution of SQF or CLF problem, since reordering the keywords may result in better solutions.

The bipartite graph algorithms can be modified for many different use cases. All hierarchical algorithms can also be modified by adding weights. As described, there are many possible variations like sorting the context data list by minimum or maximum degree. The number of iterations k also has a big impact on the result. Another possible optimization is the pre-sorting by weighting the x_i with maximum $|N(x_i)|$ and minimal $D \setminus R$.

This paper has underlined the importance of finding the computational core of NLP problems. We have managed to find a Minimum Set-Cover reformulation of SQF and TF which lead to an accurate solving of both on real-world data. The current study was unable to reproduce this success on random input data. Thus it is recommend that further research should be undertaken to examine the impact of keyword (or descriptive elements) distributions on documents. Nevertheless these results have been very encouraging to integrate this feature in SCAIView and to do further research on the optimization and extension of this heuristic.

References

[1] J. Dörpinghaus, J. Darms, and M. Jacobs, "What was the question? a systematization of information retrieval and nlp problems." in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2018.

[2] N. R. Coordinators, "Database resources of the national center for biotechnology information," *Nucleic acids research*, vol. 45, no. Database issue, p. D12, 2017.

[3] D. Suryanarayana, S. M. Hussain, P. Kanakam, and S. Gupta, "Natural language query to formal syntax for querying semantic web documents," in *Progress in Advanced Computing and Intelligent Engineering*. Springer, 2018, pp. 631–637.

[4] D. Melo, I. P. Rodrigues, and V. B. Nogueira, "Semantic web search through natural language dialogues," in *Innovations, Developments, and Applications of Semantic Web and Information Systems*. IGI Global, 2018, pp. 329–349.

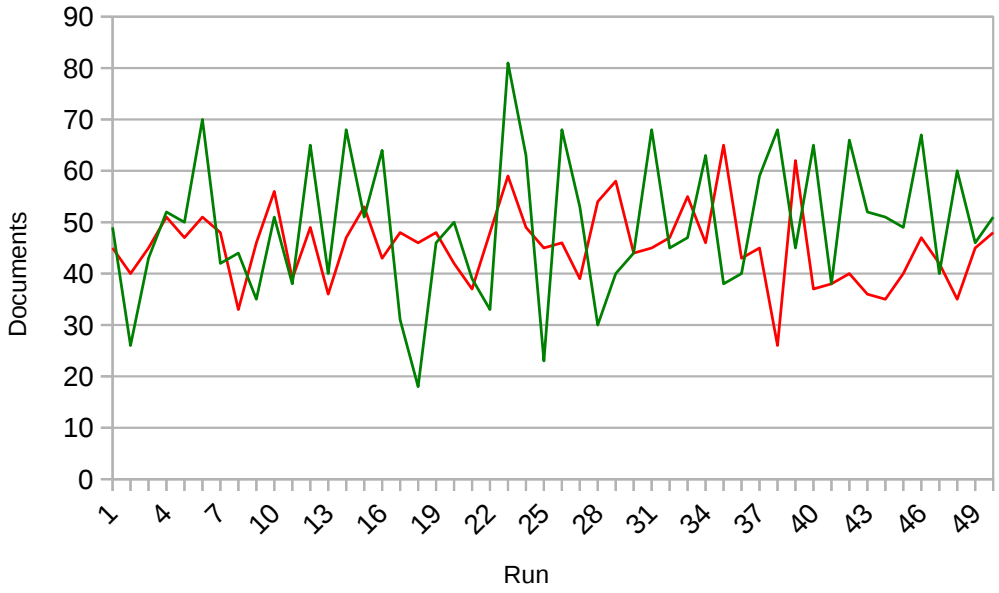

Fig. 5: Output of 50 random example runs and the number of retrieved documents in the artificial random scenario for algorithms Tree-Description (green) and Hierarchical Bipartite Cover-Description (red). The total number of documents was 150, and the document subset contains 20 documents. The number of keywords was 40. The number of iterations is $N = 4$.

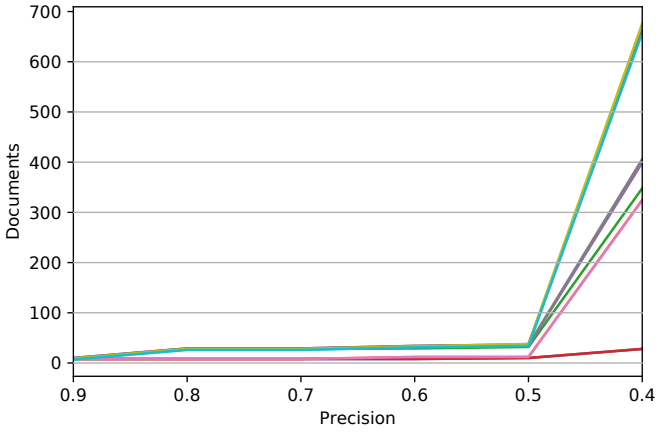

Fig. 6: Output of 10 random example runs with $|R| = 10$ on PubMed. The precision was iterated from 0.9 to 0.4. The output scales very well and is quite stable till precision 0.5.

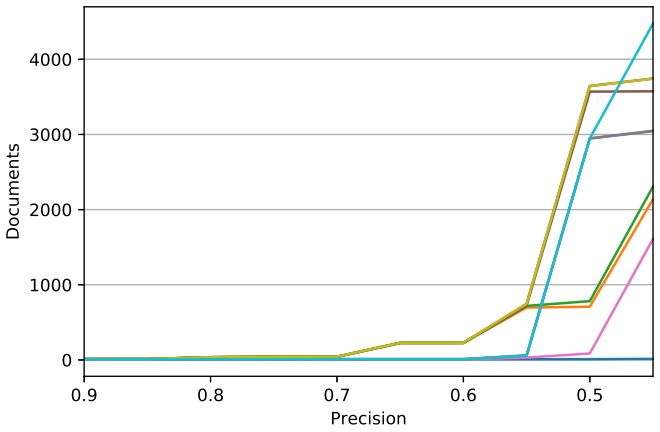

Fig. 7: Output of 10 random example runs with $|R| = 10$ on PubMed. The precision was iterated from 0.9 to 0.4. The output scales very well and is quite stable till precision 0.5.

[5] J. Lin and W. J. Wilbur, "Pubmed related articles: a probabilistic topic-based model for content similarity," *BMC bioinformatics*, vol. 8, no. 1, p. 423, 2007.

[6] D. Newman, S. Karimi, and L. Cavedon, "Using topic models to interpret medline's medical subject headings," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2009, pp. 270–279.

[7] D. Trieschnigg, P. Pezik, V. Lee, F. De Jong, W. Kraaij, and D. Rebholz-Schuhmann, "Mesh up: effective mesh text classification for improved document retrieval," *Bioinformatics*, vol. 25, no. 11, pp. 1412–1418, 2009.

[8] Z. Lu, W. J. Wilbur, J. R. McEntyre, A. Iskhakov, and L. Szilagyi, "Finding query suggestions for pubmed," in *AMIA Annual Symposium Proceedings*, vol. 2009. American Medical Informatics Association, 2009, p. 396.

[9] A. A. Bertossi, "Dominating sets for split and bipartite graphs," *Information processing letters*, vol. 19, no. 1, pp. 37–40, 1984.
[10] M. Yannakakis and F. Gavril, "Edge dominating sets in graphs," *SIAM Journal on Applied Mathematics*, vol. 38, no. 3, pp. 364–372, 1980.
[11] B. Korte, J. Vygen, B. Korte, and J. Vygen, *Combinatorial optimization*. Springer, 2012, vol. 2.
[12] P. Camerini, G. Galbiati, and F. Maffioli, "Complexity of spanning tree problems: Part i," *European Journal of Operational Research*, vol. 5, no. 5, pp. 346 – 352, 1980. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0377221780901642
[13] E. Balas and M. W. Padberg, "On the set-covering problem," *Operations Research*, vol. 20, no. 6, pp. 1152–1161, 1972.
[14] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.