Theoretical Computer Science 836 (2020) 65–75

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A semantic relatedness preserved subset extraction method for language corpora based on pseudo-Boolean optimization

Luobing Dong [a,*], Qiumin Guo [b], Weili Wu [c], Meghana N. Satpute [d]

[a] Xidian University, Xi'an, China
[b] Beijing University of Chemical Technology, Beijing, China
[c] University of Texas at Dallas, Dallas, USA
[d] The University of Texas at Dallas, India

ARTICLE INFO

Article history:
Received 6 May 2020
Received in revised form 16 June 2020
Accepted 20 July 2020
Available online 27 July 2020

Keywords:
Semantic relatedness
Subset extraction
Language intelligence
PseudoBoolean optimization
Discrete Lagrangian method

ABSTRACT

As language corpora have been playing an increasingly important role in the field of Artificial Intelligence (**AI**) research, lots of extremely large corpora are created. However, a larger corpora size not only increases power and accuracy but also brings redundancy. Therefore, researchers began to emphasize the study of appropriate subset extraction methods. Due to the trade-off between data sufficiency and redundancy, a group of interesting and challenging problems are emerged that are studied in this paper: (1) How to make the resulting subset include as much data as possible under some necessary constraints? (2) How to preserve the potential useful semantic relatedness included in the original corpora while reducing the size of the corpora? For these two problems, existing work mainly focuses on the methods to construct particular subsets for special usage. These methods are limited in their focus. In this paper, we try to address the problems listed above. First, considering the cubic and binary semantic relatedness among tokens, we construct a general system model and formulate the mix problem as a cubic pseudo-Boolean optimization problem. Then, by analyzing the characteristics of the objective function, we transfer the problem into the maximum flow problem of a corresponding graph. Third, we propose a new algorithm by introducing discrete Lagrangian iteration method. We prove that the objective function is supermodular, which allows us to use fast minimum cut algorithms in each iteration step to propose another fast algorithm. Finally, we experimentally validate our new algorithms on several randomly created corpora.

1. Introduction

In the research community, attention and energy poured into Artificial Intelligence (**AI**) have been steadily increasing in recent years. Many breakthroughs have been made on **AI** research. And **AI** research has made great progress in replicating natural language patterns. At the same time, more and more new applications based on **AI** continue to emerge in various industries including banking, recruitment, health-care, agriculture, transit, etc. The content of **AI** research focuses on how to express, acquire, and use knowledge. Knowledge about natural language is of most importance. Therefore, the study of language intelligence has a great influence on the development of **AI**. A review of the developmental history of language

* Corresponding author.
E-mail addresses: lbdong@xidian.edu.cn (L. Dong), qmguo@mail.buct.edu.cn (Q. Guo), weiliwu@utdallas.edu (W. Wu), mns086000@utdallas.edu (M.N. Satpute).

https://doi.org/10.1016/j.tcs.2020.07.020
0304-3975/

intelligence shows that the establishment of language corpora with sufficient data is very important to the research and training of new language intelligence algorithms.

A corpus is a collection of linguistic data such as written, spoken, signed, etc. It is always analyzed in various ways to establish patterns of grammar and vocabulary usage [1,2]. Based on the analysis, AI algorithm can understand the semantics of the human language [3,4] or describe its own thinking like a human [5]. The orthodox view on corpora size is that larger size corpora always provide larger coverage of general language uses. Therefore, people hold that the larger the corpora they use are, the higher the accuracy of the AI algorithms is. Some experiments also proved this point [6]. Because of this, more and more very large corpora have been established. Some corpora have billions of words, such as iWeb corpora (14 billion words), NOW corpora (5.9 billion words), etc. Many existing corpora are still expanding. For example, the English-Norwegian Parallel Corpus (**ENPC**) was built 20 years ago. But the builders believed that the small parallel corpora may be questioned, and they recently expanded **ENPC** into the **ENPC+** which has three times the size of the fiction part of the original ENPC [7].

However, a larger corpora size not only increases power and accuracy but also brings redundancy and noise [8]. As we all know, the complexity of **AI** algorithms always depends on the size of their search space which is usually proportional to the size of the corpora that they use [9]. Obviously, this redundancy and noise will bring some difficulty to the training of new algorithms, and will even lower their efficiency [8]. Experiments suggested that more data (individual style, genre, co-occurrence, semantic relatedness, etc.) in larger corpora may have a greater impact on the accuracy of the results of **AI** algorithms instead of mere corpora size [7].

In fact, we should remember that not even a very big corpus can include all varieties of a language. On the other hand, a corpus with appropriate size that only contains the kind of sample you need will be more convenient to use. For example, if you want to learn some special modal verbs in detail, a small corpus with restricted samples will be better for you than BNC which includes about 250,000 occurrences of the modal 'will' alone. For the above reasons, researchers have begun to pay attention to the study of large corpus subset extraction methods [10,9,11,12,8]. However, existing works mainly focus on the methods to construct particular subsets for special usage. These methods have limited applicability [11,10]. [8] proposed a general method to extract subset which has most data under some constraints. However, this method can only work when the objective function is modular and the constraint function is submodular. [9] proposed some methods to handle the problem when objective and constraint functions are submodular.

To the best of our knowledge, all existing work does not consider semantic relatedness preservation when extracting a subset of large corpora. The meaning of words and the semantics of phrases are always included in the context. For example, "[stock] in a business" implies the financial sense, but "[stock] in a bodega" is more likely to refer to goods on the shelves of a store [13]. It is easy for humans to distinguish the meaning of the words in the text, because we have a lot of common sense knowledge about how the world works and how it relates to language. But machines can not do this easily. Therefore, we should preserve as much semantic relatedness among tokens as possible.

In this paper, we address the problem of how to extract a subset that includes as much data as possible under some necessary constraints from large corpora. The contributions of this paper are as follows:

- We introduce the semantic relatedness preservation into subset extraction process for the first time.
- Considering the ternary and binary semantic relatedness among tokens, we construct a general system model and formulate the objective function as a cubic pseudo-boolean optimization problem.
- We show that the problem is NP-hard and the objective pseudo-boolean function is supermodular. We constructed an equivalent graph of the system. By analyzing the characteristics of the objective function, we transfer the subset extraction problem into the maximum flow problem of the equivalent graph.
- We introduce the discrete Lagrangian iteration method and propose a general algorithm for subset extraction from large corpora. By generating the equivalent graph of the objective function and accelerating each iteration step, we propose another new fast algorithm based on the minimum cut method.

This paper is organized as follows. In Section 2, we review some theories for solving the objective functions. We construct a general system model and formulate the objective function, in Section 3. In Section 4, we analyze the NP-hardness and supermodularity of the objective function. The equivalent graph is introduced too. In Section 5, we propose a new algorithm by introducing discrete Lagrangian iteration method. We try to accelerate each iteration step and then propose another fast algorithm. In Section 6, we conduct experiments to evaluate the accuracy of our new algorithms.

2. Related work

We present in this section challenges and related works that are linked with the problem of semantics relatedness preservation in large corpora subset extraction (**SRPCSE**). We also provide necessary background and definitions that are original contributions of this work.

*2.1. Challenges for **SRPCSE***

As we mentioned above, while they can provide more accuracy, very large language corpora also present some serious problems for related novel **AI** algorithm researches. Researchers always want to be able to test the correctness of the

new algorithm and its actual effects as quickly as possible. Because a quick test means that they can spend more time on handling potential problems of the novel algorithm and more time on improving its performance, instead of spending enormous time on optimizing the test project. On the other hand, more noise and redundancy in larger corpora will also affect the efficiency and accuracy of the new algorithm.

One way to address those above problems is to produce a special smaller version of the corpus, and the another way to do this is to draw a subset uniformly under some constraints. The former method is the most common one. Most current large corpora are divided into multiple sub-corpora. For example, at least three sub-corpora are built for the famous GNOME corpus: the museum sub-corpus, the pharmaceutical sub-corpus and the tutorial sub-corpus [14]. The I3media corpus has 6 sub-corpora in [15]. British National Corpus even provides tools for users to define their own sub-corpus [16]. Sub-corpus topic modeling (**STM**) is another sub-corpus building method [17]. [8,9] both modeled corpora subset extraction problem. [8] considered the problem to create a corpus of spontaneous conversational speech with limited (small) vocabulary. The authors expressed it as an optimization problem over a submodular function and converted their submodular minimization problem to the problem of finding minimum s-t cuts in a graph. [9] formulated the problem of selecting a high-quality, limited complexity sub-corpus as four different submodular functions optimization problems. Existing works hardly consider semantic relatedness preservation which is very important for language intelligence research.

The main purpose of semantic relatedness measurement is to allow computers to reason about written text [18]. It is widely used in language intelligence related **AI** researches. Many different methods are proposed to measure semantic relatedness between tokens [19–21]. In a sense, the advantage of large corpus is that it has plenty of semantic relatedness among its tokens. However, deletion of the tokens in the process of subset extraction tends to remove semantic relatedness. Therefore, how to preserve the potential useful semantic relatedness included in the original corpora while reducing the size of the corpora becomes a challenge that we must face to.

2.2. Pseudoboolean function and supermodular

Set function, i.e., the real mapping from subsets of a finite set $S = (s_1, s_2, \cdots, s_m)$ to real numbers is a general tool to solve **SRPCSE** problem [8,9]. A set function $f : 2^{|S|} \to R$ can be interpreted as a mapping from binary vectors to real numbers if we replace the finite set S with its characteristic vector. This mapping is called a pseudoboolean function [22]. A pseudoboolean function can be expressed as a function of m binary variables $(x_1, x_2, \cdots, x_m)$ [3]:

$$f(x_1, x_2, \cdots, x_m) = \sum_{i=1}^{l} a_i \prod_{j \in N_i} x_j + k \tag{1}$$

where $a_i (1 \le i \le l)$ is a real number and $a_i \neq 0$. N_i is the index subset of the variables in the ith monomial, and k is a constant. The m-vector $x \in \{0, 1\}^m$ is the characteristic vector of the subset $S_x = \{i | i \in E, x_i = 1\}$, $E = \{1, 2, \cdots, m\}$, and $l \le 2^{|E|}$.

A set function $f : 2^{|S|} \to R$ is supermodular if $f(A \cup \{j\}) - f(A) \le f(B \cup \{j\}) - f(B)$ for all $A \subseteq B \subseteq S$. For any $A \subseteq S$, $j \notin A$, $f_A(j) = f(A \cup \{j\}) - f(A)$ is the marginal contribution of element j with respect to set A. Intuitively, supermodularity means that the marginal contribution of all $j \in S$ does not decrease as the size of the set increases.

The optimization of an arbitrary pseudoboolean function belongs to the class of the so-called NP-complete problems [23]. But researchers found that the optimization of a special subclass of a pseudoboolean function can be solved in polynomial time. The special subclass is the so-called negative-positive pseudoboolean function. We rewrite a pseudo-boolean function $f(x_1, x_2, \cdots, x_m)$ as: $f(x_1, \cdots, x_m) = x_i x_j \varphi_{ij}(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_{j-1}, x_{j+1}, \cdots, x_m) + \psi_{ij}(x_1, \cdots, x_m)$ where the monomials of ψ_{ij} do not contain the product $x_i x_j$. f is supermodular if and only if the second derivative φ_{ij} is nonnegative whatever the values assigned to the other variables [23]. The maximization of pseudoboolean functions with nonnegative coefficients of higher degree (more than one) terms which is so-called negative-positive pseudo-boolean function can be converted into a maximum network flow computation in an associated graph [24]. Therefore, maximizing supermodular pseudoboolean functions can be performed polynomially via minimum cut computations.

Obviously, the nonnegative condition above is difficult to be satisfied. [23] found that the whole set of cubic supermodular pseudoboolean functions can be maximized by the above minimum cut method. A cubic pseudoboolean function $f(x_1, x_2, \cdots, x_m)$ is supermodular if and only if it can be written as formula (2) and satisfies the condition represented by formula (3). Here, J^- represents the index set of the linear terms which have negative coefficients. J^+ represents the index set of the linear terms which have positive coefficients.

$$\begin{aligned} f(x_1, x_2, \cdots, x_m) = &- \sum_{j \in J^-} c_j x_j + \sum_{j \in J^+} c_j x_j \\ &+ \sum_{i \in I, j \in J} c_{ij} x_i x_j + \sum_{i \in I, j \in J, k \in K^+} c_{ijk} x_i x_j x_k \\ &- \sum_{i \in I, j \in J, k \in K^-} c_{ijk} x_i x_j x_k + k \end{aligned} \tag{2}$$

$$c_j \geq 0, \quad c_{ij} \geq 0, \quad c_{ijk} \geq 0 \quad and \quad c_{ij} \geq \sum_{i \in I, j \in J, k \in K^-} c_{ijk} \tag{3}$$

2.3. Discrete Lagrangian method

Traditionally, the Lagrangian method is used to solve continuous constrained optimization problems. When we maximize a continuous constrained function, we can find the optimal solution by doing descents in the Lagrange-multiplier space and ascents in the original variable space until a equilibrium is reached. [25] extended continuous Lagrangian method so that it can be used to solve discrete constrained optimization problems. They defined a new gradient operator that can work in discrete space.

Discrete Lagrangian function F is defined as:

$$F(x, \lambda) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x) \tag{4}$$

where $x = (x_1, x_2, \cdots, x_m)^T \in \mathbb{Z}^m$, and $\lambda = (\lambda_1, \cdots, \lambda_n)^T \in \mathbb{R}^n$ is the vector of the Lagrange multipliers. For every multi-variable continuous function, its gradient has a component for each direction which represents the rate of change in this direction of this function. To a discrete function, there is a similar symbol which represents the change rate of the function in each direction. The difference gradient operator $\triangle_x F(x, \lambda) = (\zeta_1, \zeta_2, \cdots, \zeta_m)^T \in \{-1, 0, 1\}^m$, $\sum_{i=1}^{m} |\zeta_i| = 1$, and at most one ζ_i is non-zero. For any x' that differs from x by at most value 1 in one dimension, i.e., $\sum_{i=1}^{m} |x'_i - x_i| = 1$, $F(x - \triangle_x F(x, \lambda), \lambda)) \geq F(x', \lambda)$. If $\forall x'$, $F(x, \lambda) \geq F(x', \lambda)$, then $\triangle_x F(x, \lambda) = 0$.

Based on this definition, Lagrangian methods for continuous problems can be extended to discrete problems:

Theorem 1 *(Shang and Wah 1998). A saddle point* (x^*, λ^*) *of formula (4) can be reached by interatively calculation* $x^{k+1} = x^k - \triangle_x F(x_k, \lambda^k)$ *and,* $\lambda^{k+1} = \lambda^k + g(x_k)$.

Here $\triangle_x F(x_k, \lambda^k)$ is the direction with the maximum gradient, k is the iteration index.

3. Problem formulation

We target for the **SRPCSE** problem, which requires as much semantic relatedness preservation as possible in the process of corpus subset extraction. The **SRPCSE** takes the original corpus token set as input, performs a sequence of comparing operations onto the tokens under some constraints, and then outputs the results. This section presents a general system model and formalizes the language and variables used throughout this paper.

In this work, we consider tokens as the basic unit of extraction. For the corpora that are not token-based, all conclusions can be easily extended to be used. We use set $T = \{t_1, t_2, \cdots, t_n\}$ to denote the token set of the original corpus. For the simplicity of discussion, we use "attribute" to represent the annotated label of each token. For example, in the BEST 2009 corpus, there are many labels (such as lengths, frequency, pronunciation, category, etc.) annotated to each token [26]. Researchers often use these labels/attributes to describe the requirement or the constraints of the objective subsets [8]. Some special annotations are especially important for special **AI** research and must be considered in the process of the corresponding extraction, such as the "emotional lexicon classification" for the emotion related **AI** research. If there are m unary attributes annotated for each token in the corpus, vector $A = (A^1, A^2, \cdots, A^m)$ represents the value sets of these m attribute sets where $A^j \in \mathbb{R}^n$. Any $a_k^j \in A_j$ represents the value of jth unary attribute of t_k. Here, the unary attribute means its value depends only on the token it belongs to. Accordingly there are binary and multivariate attributes. We use unary and binary attributes to describe constraints. We use matrix $B = (B^1, B^2, \cdots, B^l)$ to denote all the value matrices of binary attributes where $B^i \in \mathbb{R}^{n \times n}$. Any $b_{kp}^i \in B^i$ represents the value of the ith binary attribute between t_k and t_p. For the semantic relatedness, we do not consider the details of its measurement method. We just use $S^3 \in \mathbb{R}^{n \times n \times n}$, $S^2 \in \mathbb{R}^{n \times n}$ and $S^1 \in \mathbb{R}^n$ to denote the cubic, binary and unary semantic relatedness values among tokens respectively. Assume that vector $X = (x_1, x_2, \cdots, x_n) \in \{0, 1\}^n$ represents the extraction result. $x_i = 1$ denotes the token t_i is in the result subset, while $x_i = 0$ denotes t_i is not in the result subset. We can define the **SRPCSE** problem as follows.

Definition 1 *(**SRPCSE** problem).* Given corpus C with token set T, unary attribute value vector A, binary attribute value vector B, and semantic relatedness value vectors S^3, S^2, S^1. **SRPCSE** problem is finding the optimal X with:

$$\arg\max_X \Big(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} s_{ijk}^3 x_i x_j x_k + \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij}^2 x_i x_j + \sum_{i=1}^{n} s_i^1 x_i \Big) \tag{5}$$

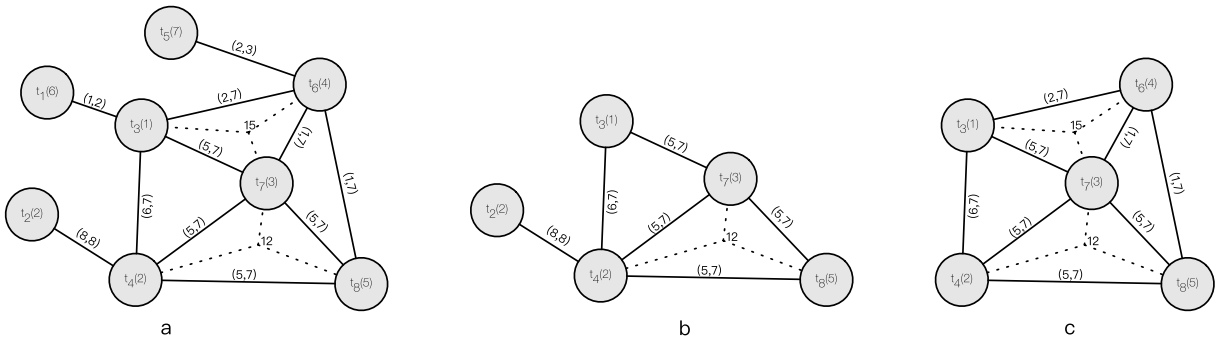

Fig. 1. Example of subset extraction. The weight of node represents the value of unary attribute (A_1), the first item of the solid line's weight denotes the value of binary attribute (B_1), the second item of the solid line's weight denotes the value of binary semantic relatedness (S_2), and the weight of dotted line represents the value of cubic semantic relatedness. a, the original corpus. b, the subset of $\arg\max_X \sum_{i=1,j=1}^{n} b_{ij}^1 x_i x_j$ s.t. $\sum_{i=1}^{n} a_i^1 \leq 15$. c, the subset of $\arg\max_X (\sum_{i=1,j=1,k=1}^{n} s_{ijk}^3 x_i x_j x_k + \sum_{i=1,j=1}^{n} s_{ij}^2 x_i x_j)$ s.t. $\sum_{i=1,j=1}^{n} b_{ij}^1 x_i x_j \leq 30$, $\sum_{i=1}^{n} a_i^1 \leq 15$.

s.t.

$$\begin{aligned} &\sum_{k=1}^{n} a_k^r x_k \leq u_r \quad u_r > 0 \\ &\sum_{k=1}^{n}\sum_{p=1}^{n} b_{kp}^l x_k x_p \leq v_l \quad v_l > 0 \end{aligned} \tag{6}$$

Note that there can be multiple constraints in formula (6) which associate to different attributes. For the simplicity of discussion, we only consider one unary attribute constraint and one binary attribute constraint in this paper. Our new algorithms can be easily extended to multi-constraints situations. We do not consider the maximization of the information related to the attributes. We can just add corresponding calculation in formula (5) if we need to do that and all the theorems and properties that are discussed in the following part are still applicable. IBM researchers found through experiments that in the n-grams models, when n is equal to 3, the generated phrases are almost impossible to appear in the actual language, let alone more than 3 [27]. Therefore, it is reasonable to consider the semantic relatedness generated from up to three tokens together.

We can also use graph like Fig. 1a to describe the **SRPCSE** problem. The nodes denote the tokens. The weight vectors of nodes represent unary attribute values and the weight vectors of edges denote the values of binary attributes and semantic relatedness. The cubic semantic relatedness values can be represented by the values of compound lines among any three nodes. Fig. 1b shows the result without considering semantic relatedness preservation and Fig. 1c shows the opposite situation.

4. Theoretical analysis

As the definition of the **SRPCSE** problem, for given attribute sets A, B and semantic relatedness sets S_i, we aim to provide the optimal subset which is denoted by X, such that the semantic relatedness contained is maximized under some constraints. The key challenge in solving the above optimization problem **SRPCSE** of formula (5) and (6) is in supermodularity and NP-hardness. In this section, we first analyze the NP-hardness and supermodularity of the objective function. Then the equivalent graph is introduced.

Theorem 2. *The optimization **SRPCSE** problem (**Definition 1**) is NP-hard.*

Proof. Consider an instance of the NP-complete Knapsack problem. Given a set of commodity $D = (d_1, d_2, \cdots, d_n)$ with weights $(w_1, w_2, \cdots, w_n)$ and profits $(p_1, p_2, \cdots, p_n)$, finding a subset of commodity whose total profit is as large as possible, and the total weight is at most b. We show that this can be viewed as a special case of **SRPCSE** problem. Given an arbitrary instance of the Knapsack problem, we define a corresponding language corpus with n tokens. Assume that there are only one unary attribute and one unary semantic relatedness associated to each token. There is a token t_i corresponding to each commodity d_i, and the semantic relatedness that it has is equal to p_i. In addition, there is an attribute value a_i of each token corresponding to w_i. The Knapsack problem is equivalent to finding subset of tokens (corresponding to X) that contains maximum semantic relatedness in this corpus with constraints $u = b$. If any subset can be obtained, then the Knapsack problem must be solvable. □

As described in the proof of Theorem 1, the problem is NP-hard even though we just consider one unary attribute constraint. It must be more difficult to solve when more complex binary constraints are added. Furthermore, there are up to $\binom{n}{3}$ cubic semantic relatednesses and $\binom{n}{2}$ binary semantic relatednesses among n tokens. Therefore, the search space will be very large. Fortunately, the objective function has the other properties described below, which we can use to simplify it.

Theorem 3. *The objective function of **SRPCSE** problem (formula (5)) and its constraint functions (formula (6)) are non-decreasing and supermodular.*

Proof. The objective function (formula (5)) and its constraint functions (formula (6)) are all sums of terms $x_i x_j x_k$, $x_i x_k$ or x_i with positive coefficients. Since they are products of 0 - 1 variables each of which is an indicator for an token to be selected into the final set or not, it is clear that this product has monotone nondecreasing property.

To see the supermodularity property, let's denote $f(A) = x_{i_1} x_{i_2} \cdots x_{i_k}$. $x_{i_k} = 1$ represents the element i_k is in the final set. Otherwise, $x_{i_k} = 0$. Then $f(A) = 0$ if $A \not\supseteq \{i_1, i_2, ..., i_k\}$, and $f(A) = 1$ if $A \supseteq \{i_1, i_2, ..., i_k\}$. Obviously, if function f is supermodular, the objective function and its constraint functions are supermodular.

Consider two sets $A \subset B$ and an element $i \notin B$, if $i \notin \{i_1, i_2, ..., i_k\}$, then $f(A) = f(A \cup \{i\})$ and $f(B) = f(B \cup \{i\})$. Hence $f(A \cup \{i\}) - f(A) = f(B \cup \{i\}) - f(B) = 0$. Hence, we may assume $i = i_1$ without loss of generality. Now, we have three cases.

Case 1. $A \supseteq \{i_2, ..., i_k\}$. We have $f(A \cup \{i_1\}) = f(B \cup \{i_1\}) = 1$ and $f(A) = f(B) = 0$. Hence $f(A \cup \{i_1\}) - f(A) = f(B \cup \{i_1\}) - f(B) = 1$.

Case 2. $A \not\supseteq \{i_2, ..., i_k\}$ and $B \supseteq \{i_2, ..., i_k\}$. We have $f(A) = f(A \cup \{i_1\}) = f(B) = 0$ and $f(B \cup \{i_1\}) = 1$. Hence $f(A \cup \{i_1\}) - f(A) = 0 < f(B \cup \{i_1\}) - f(B) = 1$.

Case 3. $B \not\supseteq \{i_2, ..., i_k\}$. We have $f(A) = F(B) = f(A \cup \{i_1\}) = f(B \cup \{i_1\}) = 0$. Hence $f(A \cup \{i_1\}) - f(A) = f(B \cup \{i_1\}) - f(B) = 0$.

Above three cases showed that f have marginal value monotone non-decreasing. Hence f is supermodular. □

Theorem 4. *Let $f(X)$ denote the objective function of formula (5), $g_1(X)$ and $g_2(X)$ denote the unary and binary constraint functions in formula (6) respectively. For any fixed $\lambda_1 > 0$, $\lambda_2 > 0$, if $f(X) - \lambda_1(g_1(X) + g_2(X)) + \lambda_2(u_1 + v_1)$ is maximized at X_i, and there exist small enough ϵ_1 and ϵ_2 such that $g_1(X_i) = u_1 - \epsilon_1$ and $g_2(X_i) = v_1 - \epsilon_2$ ($\epsilon_1 \geq 0$ and $\epsilon_2 \geq 0$), then X_i is an optimal solution for the **SRPCSE** problem.*

Proof. Let $H(X) = f(X) - \lambda_1(g_1(X) + g_2(X)) + \lambda_2(u_1 + v_1)$. Assume that X_i is not an optimal solution for formula (5) and (6). There must exist X_j, such that $f(X_j) > f(X_i)$, $g_1(X_j) \leq u$, and $g_2(X_i) \leq v$. According to **Theorem 3**, $H(X)$ is non-decreasing. Then we can get that $H(X_j) > H(X_i)$. This is a contradiction to the pre-set condition that $H(X)$ is maximized at X_i. □

Because of the NP-hardness of the **SRPCSE** problem (**Theorem 2**), we hardly find a polynomial time algorithm for it. However, we can define the equivalent Lagrangian function $Q(X)$ (formula (7)) of the objective function according to **Theorem 4** and Envelope Theorem.

$$\begin{aligned} Q(X) = & \\ & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} s^3_{ijk} x_i x_j x_k + \sum_{i=1}^{n}\sum_{j=1}^{n} s^2_{ij} x_i x_j + \sum_{i=1}^{n} s^1_i x_i \\ & -\lambda_1(\sum_{k=1}^{n} a^r_k x_k - u_r) - \lambda_2(\sum_{k=1}^{n}\sum_{p=1}^{n} b^l_{kp} x_k x_p - v_l) + \epsilon \end{aligned} \tag{7}$$

We can just delete the positive linear term because the corresponding x_i must be 1. Obviously, $Q(X)$ is supermodular if $s^2_{ij} \geq \lambda_2 b^l_{ij}$. We suppose this condition is always valid. Because the semantic relatedness associated to x_i and x_j is usually small compared to their contribution to the binary constraint when $s^2_{ij} < \lambda_2 b^l_{ij}$. At this time, we can just delete the binary and cubic terms in $Q(X)$ which contain both x_i and x_j to reduce the probability that these two tokens are selected. By replacing x_i by $\bar{x_i} = 1 - x_i$ in all the negative terms of formula (7), we can get rid of the negative coefficients. Then $Q(X)$ can be changed as:

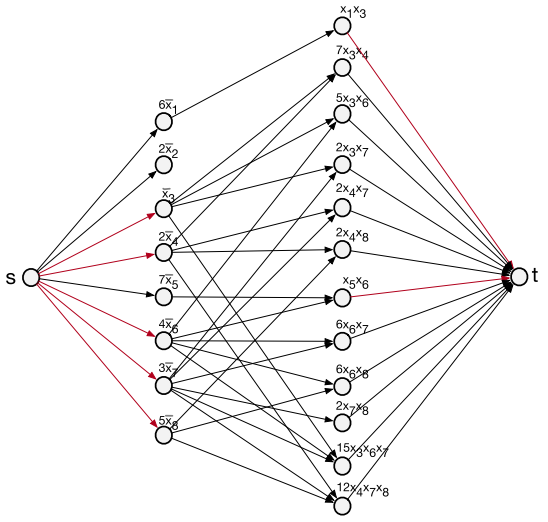

Fig. 2. Graph corresponding to the problem of Fig. 1c. Red lines represent the minimum cut. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\begin{aligned} Q(X) = &\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} s_{ijk}^{3} x_i x_j x_k \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{n}(s_{ij}^{2} - \lambda_2 b_{ij}^{l}) x_i x_j + \lambda_1 \sum_{k=1}^{n} a_k^{r} \bar{x}_k \\ &-\lambda_1 \sum_{k=1}^{n} a_k^{r} + \lambda_1 u_r + \lambda_2 v_l + \epsilon \end{aligned} \tag{8}$$

We assume that there are $e \le 2n$ linear terms and $f \le (n^3 + 2n^2)$ nonlinear terms in formula (8). Using the following process we can get the equivalent conflict graph G of $Q(X)$.

- Setting two sets of nodes $\Gamma = (\gamma_1, \gamma_2, \cdots, \gamma_e)$ and $\Upsilon = (\upsilon_1, \upsilon_2, \cdots, \upsilon_f)$. Each node γ_i (υ_j) is corresponding to a term in formula (8) which includes original (complement) variable. The weight of each node equals to the coefficient of its corresponding term.
- Adding edges with upper capacity bound $+\infty$ between nodes in Γ and nodes in Υ. There exists an edge between γ_i and υ_j if and only if the associated term of γ_i contains a variable and that the associated term of υ_j contains the same variable complemented.
- Adding a node s and linking it to each node in Γ. The edge between s and γ_i has the upper capacity bound the weight of γ_i.
- Adding a node t and linking each node in Υ to t. The edge between υ_j and t has the upper capacity bound the weight of υ_j.

Billionnet and Minoux proved that maximizing a cubic supermodular pseudoboolean function can be solved polynomially by a maximum flow algorithm [23]:

Theorem 5 *(Billionnet and Minoux 1985). The problem of maximizing a cubic supermodular pseudoboolean function can be seen as a special case of the maximum weight stable set problem in a bipartite graph, hence can be solved polynomially by a maximum flow algorithm.*

When $s_{ij}^2 \ge \lambda_2 b_{kp}^l$, condition of formula (8) is satisfied, thus $Q(X)$ is a cubic supermodular pseudoboolean function. Then the maximization of $Q(X)$ can be reformulated as the search for minimum cut in graph G. Obviously, there is no K_5 and $K_{3,3}$ in graph G. Thus, G is a planar graph. This means the maximization of $Q(X)$ for fixed λ_1 and λ_2 can be calculated in $O((e+f)\log\log(e+f))$ time [28]. We reconsider about the problem of Fig. 1c, when $\lambda_1 = \lambda_2 = 1$, $Q(X) = 15x_3x_6x_7 + 12x_4x_7x_8 + x_1x_3 + x_3x_4 + 5x_3x_6 + 2x_3x_7 + 2x_4x_7 + 2x_4x_8 + x_5x_6 + 6x_6x_7 + 6x_6x_8 + 2x_7x_8 + 6\bar{x}_1 + 2\bar{x}_2 + \bar{x}_3 + 2\bar{x}_4 + 7\bar{x}_5 + 4\bar{x}_6 + 3\bar{x}_7 + 5\bar{x}_8 + 15$. Fig. 2 shows the equivalent graph. The red lines in Fig. 2 represent the minimum cut. And the optimal $X = (0, 0, 1, 1, 0, 1, 1, 1)$. We should note that the minimum cut of graph G can just give the maximization of $Q(X)$ which may not guarantee the constraints of formula (6). We will consider this issue in the next section.

5. Algorithms for the SRPCSE problem

In this section, we propose two algorithms to solve the **SRPCSE** problem. Firstly, we introduce a discrete Lagrangian iteration method based algorithm (**DLIB** algorithm) to get the optimal solution of formula (5). Then we try to use the equivalence between the maximization of $Q(X)$ and the minimum cut of graph G to construct another algorithm (**MQMCG** algorithm).

5.1. *DLIB* algorithm

As we mentioned in the front section, the discrete Lagrangian method can be used to solve discrete constrained optimization problems. We introduce this method into the **SRPCSE** problem by replacing the objective function (formula (5)) and constraint functions (formula (6)) with the Lagrangian function of formula (7). We then calculate the optimal value of the objective variable X and the Lagrangian-multiplier λ_i by iteratively doing descents in the Lagrange-multiplier space and ascents in the objective variable space. X^{h+1}, λ_1^{h+1} and λ_1^{h+1} in each iteration can be obtained by formula (9). Obviously, the value of $\triangle_X Q(X^h, \lambda_1^h, \lambda_2^h)$ is very important for the calculating of the above three variables. But, as we all know, the value of $\triangle_X Q(X^h, \lambda_1^h, \lambda_2^h)$ is not unique. We choose the first one that reduces $Q(X)$ (the so-called hill climbing method [25]). Furthermore, instead of updating λ_1 and λ_2 in each iteration, we update them when a local optimal X of $Q(X)$ is reached.

$$\begin{aligned} X^{h+1} &= X^h + \triangle_X Q(X^h, \lambda_1^h, \lambda_2^h) \\ \lambda_1^{h+1} &= \lambda_1^h + \sum_{k=1}^{n} a_k^r x_k^h - u_r \\ \lambda_2^{h+1} &= \lambda_2^h + \sum_{k=1}^{n}\sum_{p=1}^{n} b_{kp}^l x_k^h x_p^h - v_l \end{aligned} \tag{9}$$

The details of the **DLIB** algorithm are shown in Algorithm 1. All necessary variables are initialized in Line 1. The local optimal X is found step by step for each λ_1 and λ_2 (Line 4 to Line 7). The parameter ζ controls the magnitude of the changes in λ_1 and λ_2 (Line 8 to Line 9). When the value of $Q(X)$ can not be changed by the iteration or the times of iteration get to the upbound (τ), the process will be over (Line 11 to Line 14).

Algorithm 1 **DLIB** Algorithm.

Input X^0, λ_1^0, λ_2^0, A^1, B^1, S^1, S^2, S^3, u_1, v_1, ϵ, ζ, τ;
1: $h = 0$, $z = 0$, $t = 0$, $flag = false$
2: **while** $flag == False$ **do**
3: $X' = X^h$
4: **while** $\triangle_X Q(X^h, \lambda_1^z, \lambda_2^z) <> 0$ **do**
5: $X^{h+1} = X^h + \triangle_X Q(X^h, \lambda_1^z, \lambda_2^z)$
6: $h = h + 1$
7: **end while**
8: $\lambda_1^{z+1} = \lambda_1^z + \zeta(\sum_{k=1}^{n} a_k^1 x_k^h - u_r)$
9: $\lambda_2^{z+1} = \lambda_2^z + \zeta(\sum_{k=1}^{n}\sum_{p=1}^{n} b_{kp}^1 x_k^h x_p^h - v_l)$
10: $z = z + 1$ $t = t + 1$
11: **if** $Q(X^h, \lambda_1^z, \lambda_2^z) == Q(X', \lambda_1^z, \lambda_2^z)$ or $t == \tau$ **then**
12: $flag = True$
13: return$(X^h, \lambda_1^z, \lambda_2^z)$ break
14: **end if**
15: **end while**

5.2. *MQMCG* algorithm

The **DLIB** algorithm will finally stop at a local optimal point or saddle point. But its efficiency could be low because of the low searching speed of the local optimal objective variable X. As is shown in Line 4 to Line 7 of Algorithm 1, one update can only make the objective variable advanced for one step in one direction. We know that the optimal point of $Q(X)$ for fixed λ_i can be reached in $O((e+f)\log\log(e+f))$ time when $Q(X)$ is supermodular. Therefore, we can replace the updating process of X^h by planar graph minimum cut process in the **DLIB** algorithm to enhance the efficiency of the algorithm. The sufficient and necessary condition of the supermodularity is $s_{ij}^2 \geq \lambda_2 b_{ij}^l$. Fortunately, this condition will always be true because there are rarely binary constraints in actual applications. If it is not satisfied, the associated tokens always have more contribution on constrained attribute than their contribution on the semantic relatedness. So we can simply delete the corresponding cubic and binary terms in $Q(X)$ to make sure $Q(X)$ is supermodular. Following the above idea, we propose the **MQMCG** algorithm (Algorithm 2).

Algorithm 2 MQMCG Algorithm.

1: Initializing variables.
2: **while** X^h is not a solution **do**
3: Constructing equivalent graph G^h of $Q(X^h, \lambda_1^h, \lambda_2^h)$
4: $m_c = Min\_Cut(G)$
5: Calculating X^{h+1} from m_c
6: Calculating λ_1^h, λ_2^h
7: **end while**

Table 1
Values of parameters in experiments.

n	p_1	p_2	Upper attribute	Upper semantic relatedness
100K	0.0005	0.0001	5	10
λ^0	u_1	ζ	Lower attribute	Lower semantic relatedness
1	3000	0.003	0	0

The variables initialization method is the same as that of Algorithm 1 (Line 1). The stop condition (Line 2) and the updating strategy of λ_i are also the same as that of Algorithm 1. We can use any one of the existing fast minimum cut algorithms to generate the *Min_Cut* function in Line 4.

The **MQMCG** algorithm can reduce the time consumption of each iteration of the **DLIB** algorithm. But it also exposes another issue that the result is the optimal point of $Q(X)$ (formula (7)) rather than that of the real objective function (formula (5)), although it can get the best balance between the requirement and the constraints. For example, in the above example of graph Fig. 1c, the result will be $X = (1, 0, 1, 1, 0, 1, 1, 1)$ if we change the weight vector of the edge between t_1 and t_3 to $(1, 8)$. This will make the constraint condition $\sum_{i=1,j=1}^{n} b_{ij}^1 x_i x_j \leq 30$ and $\sum_{i=1}^{n} a_i^1 \leq 15$ false. Sometimes the constraint condition is strict while sometimes this above balance is acceptable. We must handle this issue if it is the former case.

Assume that S^h represents the result subset corresponding to X^h. If it contains more amount of constrained attributes values, we cyclically remove from S^h the point that has the greatest relative contribution to the constraint attributes until the constraint conditions are true. We measure the relative contribution of a token t_i to the constraint attributes by the θ_i that is defined in formula (10).

$$\theta_i = \frac{\sum_{j\in\mathbb{N}} b_{ij} + a_i}{\sum_{j\in\mathbb{N}} b_{ij} + a_i + \sum_{j,k\in\mathbb{N}} s_{ijk}^3 + \sum_{j\in\mathbb{N}} s_{ij}^2 + s_i^1} \tag{10}$$

Due to the space limitation, we do not give the detail description of the removing process. We can either use this process after the *Min_Cut* operation in each iteration or just do it one time when the whole loop of the **MQMCG** algorithm is over.

6. Experimental results

In this section, we evaluate our algorithms and compare it against the state-of-the-art greedy algorithm. To evaluate the subset extraction algorithms, we randomly generate a corpus and consider the following parameters: corpus size (n), the probability of semantic relatedness among arbitrary tokens (p_1), and the probability of the existence of binary attribute between any two tokens (p_2). We compare the total semantic relatedness preserved in the result subset. The values of parameters we set in experiments are shown in Table 1. In practical applications, we can always limit the sizes of connected subsets to less than $100K$ by eliminating some weak semantic relatedness links. Therefore, we set the size of corpus as $100K$. We use Mozes's minimum cut algorithm [28] which is the fastest algorithms currently known in the **MQMCG** algorithm.

We perform 1000 experiments on each corpus and take the average of results. Table 2 shows one simulation result of the **MQMCG** algorithm and the greedy algorithm (the initial subset is $\vec{0}$), and five results of the **DLIB** algorithm (different randomly selected initial subsets). Obviously, the **MQMCG** algorithm can preserve more semantic relatedness than the other two under the same constraints. The results of the **DLIB** algorithm depend on the initial subset and the magnitude of the changes in λ, which makes the result unstable. In experiments, ζ has a significant impact on λ. A bigger ζ will make the two new algorithms stop without iteration on λ. Conversely, a lower ζ will cause the two new algorithms to iterate too many times on λ and make them time-consuming.

7. Conclusions

In this paper we propose two Pseudo-Boolean optimization based algorithms (the **DLIB** algorithm and the **MQMCG** algorithm) by introducing the discrete Lagrangian method and the supermodular cubic Pseudo-Boolean optimization method. To make sure that the **MQMCG** algorithm can reach the optimal point of the real objective function, we define the measurement method of the relative contribution of a token. Finally, we evaluate their performance. There is none existent corpus

Table 2
Some results of algorithms.

Algorithm	Total semantic relatedness	Total attribute	λ
DLIB	3057	1725	1
	7259	3547	0.84
	6428	2964	0.82
	8537	2844	0.66
	7375	2857	0.58
MQMCG	8993	2998	0.83
Greedy	8598	2999	1

that includes labels of relatedness, we can just do some simulation experiments. Therefore, the experimental results and conclusions part are a little weak in this paper. In the future, we will try to modify some corpus by annotating the relatedness among its tokens. Then We will further improve the accuracy of the **MQMCG** algorithm when the objective function is not supermodular and do more experiments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is partly supported by National Science Foundation under Grant 1747818, and the Fundamental Research Funds for the Central Universities (JB161004).

References

[1] T.-H. Yen, J.-C. Wu, J. Chang, J. Boisson, J. Chang, Writeahead: mining grammar patterns in corpora for assisted writing, in: Proceedings of ACL-IJCNLP 2015 System Demonstrations, 2015, pp. 139–144.
[2] D. Miller, D. Biber, Evaluating reliability in quantitative vocabulary studies: the influence of corpus design and composition, Int. J. Corpus Linguist. 20 (1) (2015) 30–53.
[3] A. Caliskan, J.J. Bryson, A. Narayanan, Semantics derived automatically from language corpora contain human-like biases, Science 356 (6334) (2017) 183–186.
[4] G. Aston, Acquiring the language of interpreters: a corpus-based approach, in: Making Way in Corpus-Based Interpreting Studies, Springer, 2018, pp. 83–96.
[5] T.-H. Wen, M. Gasic, D. Kim, N. Mrksic, P.-H. Su, D. Vandyke, S. Young, Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking, arXiv preprint, arXiv:1508.01755.
[6] S.A. Crossley, M. Dascalu, D.S. McNamarac, How important is size? An investigation of corpus size and meaning in both latent semantic analysis and latent Dirichlet allocation, in: 30th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, AAAI Press, 2017.
[7] S.O. Ebeling, Does corpus size matter? Revisiting ENPC case studies with an extended version of the corpus, Nord. J. Engl. Stud. 15 (3) (2016) 33–54.
[8] H. Lin, J. Bilmes, Optimal selection of limited vocabulary speech corpora, in: Twelfth Annual Conference of the International Speech Communication Association, 2011.
[9] Y. Liu, R. Iyer, K. Kirchhoff, J. Bilmes, Svitchboard ii and fisver I: high-quality limited-complexity corpora of conversational English speech, in: Sixteenth Annual Conference of the International Speech Communication Association, 2015.
[10] D. Cutting, J. Kupiec, J. Pedersen, P. Sibun, A practical part-of-speech tagger, in: Proceedings of the Third Conference on Applied Natural Language Processing, Association for Computational Linguistics, 1992, pp. 133–140.
[11] P. Agarwal, J. Strötgen, L. Del Corro, J. Hoffart, G. Weikum, diaNED: time-aware named entity disambiguation for diachronic corpora, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, 2018, pp. 686–693.
[12] M.A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: Proceedings of the 14th Conference on Computational Linguistics-Volume 2, Association for Computational Linguistics, 1992, pp. 539–545.
[13] C. Evans, D. Yuan, A large corpus for supervised word-sense disambiguation, https://ai.googleblog.com/2017/01/a-large-corpus-for-supervised-word.html, 2017.
[14] M. Poesio, Discourse annotation and semantic annotation in the gnome corpus, in: Proceedings of the 2004 ACL Workshop on Discourse Annotation, Association for Computational Linguistics, 2004, pp. 72–79.
[15] J.M. Garrido, Y. Laplaza, M. Marquina, A. Pearman, J.G. Escalada, M.Á.R. Crespo, A. Armenta, The I3MEDIA speech database: a trilingual annotated corpus for the analysis and synthesis of emotional speech, in: LREC, Citeseer, 2012, pp. 1197–1202.
[16] L. Bowker, J. Pearson, Working with Specialized Language: A Practical Guide to Using Corpora, Routledge, 2002.
[17] T.R. Tangherlini, P. Leonard, Trawling in the sea of the great unread: sub-corpus topic modeling and humanities research, Poetics 41 (6) (2013) 725–749.
[18] I.H. Witten, D.N. Milne, An Effective, Low-cost Measure of Semantic Re-latedness Obtained from Wikipedia Links, IAAA Press, 2008.
[19] E.-Y. Ran, D. Yanay, Methods and systems of supervised learning of semantic relatedness, uS Patent 8,909,648, Dec. 9, 2014.
[20] I. Hulpuş, N. Prangnawarat, C. Hayes, Path-based semantic relatedness on linked data and its use to word and entity disambiguation, in: International Semantic Web Conference, Springer, 2015, pp. 442–457.
[21] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, R. Zamparelli, Semeval-2014 task 1: evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 1–8.
[22] E. Boros, A. Gruber, On quadratization of pseudo-Boolean functions, arXiv preprint, arXiv:1404.6538.
[23] A. Billionnet, M. Minoux, Maximizing a supermodular pseudoboolean function: a polynomial algorithm for supermodular cubic functions, Discrete Appl. Math. 12 (1) (1985) 1–11.

[24] J. Rhys, A selection problem of shared fixed costs and network flows, Manag. Sci. 17 (3) (1970) 200–207.
[25] Y. Shang, B.W. Wah, A discrete Lagrangian-based global-search method for solving satisfiability problems, J. Glob. Optim. 12 (1) (1998) 61–99.
[26] M. Boriboon, K. Kriengket, P. Chootrakool, S. Phaholphinyo, S. Purodakananda, T. Thanakulwarapas, K. Kosawat, Best corpus development and analysis, in: 2009 International Conference on Asian Language Processing, IEEE, 2009, pp. 322–327.
[27] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, J.C. Lai, Class-based n-gram models of natural language, Comput. Linguist. 18 (4) (1992) 467–479.
[28] S. Mozes, K. Nikolaev, Y. Nussbaum, O. Weimann, Minimum cut of directed planar graphs in $o(n \log\log n)$ time, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 477–494.