Abstract

Selecting efficiently a minimum amount of text from a large-scale text corpus to achieve a maximum coverage of certain units is an important problem in spoken language processing area. In this paper, the above text selection problem is first formulated as a maximum coverage problem with a Knapsack constraint (MCK). An efficient rank-predicted pseudo-greedy approach is then proposed to solve this problem. Experiments on a Chinese text selection task are conducted to verify the efficiency of the proposed approach. Experimental results show that our approach can improve significantly the text selection speed yet without sacrificing the coverage score compared with traditional greedy approach.

Index Terms: text selection, greedy approach, pseudo-greedy approach, maximum coverage problem.

1. Introduction

Selecting efficiently a minimum amount of text from a large-scale text corpus to optimize an objective function is an important problem in spoken language processing (SLP) area (e.g., [15, 14, 8, 13, 3, 5, 7]). Achieving a maximum coverage of certain coverage units is probably the most popular objective function adopted in different SLP applications (e.g., [15, 14, 8, 7]), which is also the one we studied in this paper. Other objective functions, which measure the dissimilarity between the distribution of coverage units in the selected text and a target distribution, are also studied in literature (e.g., [14, 3, 7]).

The above text selection problem can be cast as a constrained combinatorial optimization problem. Several greedy approaches were studied in literature to solve the above problem. For example, a greedy approach was proposed in [15], which begins with an empty set, then in each iteration selects a sentence giving the highest increase of the objective function to grow the set of selected text, until a termination condition is satisfied. On the contrary, the reduction approach studied in [13] begins by including all the sentences in the back-end text corpus into the set of selected text. Then the poorest sentence is selected and removed repeatedly until a termination condition is satisfied. Furthermore, a pair-exchange approach investigated in [8] begins with a set of sentences selected randomly from the back-end text corpus. Then an unexamined sentence in the back-end text corpus is exchanged temporarily with a sentence in the set of selected text. If a better objective function score is obtained, the exchange is kept; otherwise exchange is not effected. Of course, a combination of the above strategies can always be used to derive a better approach for text selection (e.g., [5, 7]).

In addition to the greedy nature of the above approaches, every sentence in the back-end text corpus has to be examined in each iteration, which is very time-consuming for a large back-end text corpus and/or a complex objective function. Therefore more efficient approaches are desirable. In this paper, we present such a new approach.

The rest of the paper is organized as follows. In Section 2, the above text selection problem is first formulated as a maximum coverage problem with a Knapsack constraint (MCK) [1]. Then in Section 3, a greedy approach is presented to solve the MCK problem (MCKP) and the upper bound of the corresponding score functions is derived. In Section 4, a more efficient rank-predicted pseudo-greedy approach is proposed to solve the MCK problem. Experiments and results are reported in Section 5. Finally, the paper is concluded in Section 6.

2. Formulation of Text Selection Problem

Let’s use \( V = \{v_i|i = 1, 2 \ldots, N_v\} \) to denote a back-end text corpus consisting of \( N_v \) sentences, and \( U = \{u_i|i = 1, 2 \ldots, N_u\} \) to denote a set of coverage units. Let’s further use \( P_t(u) \) to denote the occurrence frequency of the coverage unit \( u \), \( S \) to denote a set of sentences selected from \( V \), and \( U(S) \) to denote the set of coverage units covered by \( S \). A coverage score can then be defined as

\[
CS(S) = \sum_{u \in U(S)} P_t(u). \tag{1}
\]

Our text selection problem can then be cast as a constrained optimization problem: select \( S \) from \( V \) to maximize \( CS(S) \) subject to the constraint \( \sum_{u \in S} a_u = N_{u,S} \), where \( a_u \) denotes the number of syllables in sentence \( u \) and \( N_{u,S} \) denotes the total number of syllables in the sentences to be selected.

Let’s use \( c_{uv} \) to denote the number of times the \( u \)-th coverage unit occurs in the \( v \)-th sentence in \( V \), and a binary variable \( x_v \) to indicate if the \( v \)-th sentence is selected or not (i.e., \( x_v = 1 \), when the \( v \)-th sentence is chosen; \( x_v = 0 \), otherwise). Then the above text selection problem can be formulated as the following MCK problem [1]:

\[
\max_{S \subseteq V} \sum_{u \in U(S)} P_t(u) \left( 1 - \prod_{v \in V} (1 - c_{uv}x_v) \right) \tag{2}
\]

s.t. \( \sum_{u \in V} a_u x_v = N_{u,S} \)

\( x_v \in \{0, 1\}, \ v \in V \).

3. Greedy Approach to Solving MCKP

MCKP is known as an NP-hard problem with both maximum coverage problem and the Knapsack problem as special cases
In this section, we present a greedy approach to solving this problem and derive an upper bound of the corresponding score function.

3.1. Procedure of Greedy Approach

If the total number of syllables in $S$ is less than $N_{syl}$, repeat the following actions:

- In the $l^{th}$ iteration, identify the sentence $v(l)$ not examined yet such that
  \[
  v(l) = \arg \max_{v \in V} \frac{Value_v}{a_v}
  \]
  where \( Value_v = \sum_{u \in U} c_{uv}P_l(u) \).

- Update \( S_u = S_u \cup \{ v(l) \} \).

3.2. Upper Bound of Greedy Approach for MCKP

The above MCKP can be reformulated as the following constrained linear programming problem [1]:

\[
\begin{align*}
\max & \quad \sum_{u \in U} P_l(u)z_u \\
\text{s.t.} & \quad \sum_{v \in V} c_{uv}x_v \geq z_u \\
& \quad \sum_{v \in V} a_v x_v = N_{syl} \\
& \quad x_v \in \{0,1\}, v \in V; 0 \leq z_u \leq 1, u \in U, \\
\end{align*}
\]

where \( z_u = 1 \) represents that $u$ is covered by $U(S)$ and \( z_u = 0 \) otherwise. Suppose $L + 1$ iterations are performed in the above greedy procedure. Let’s use $U_l$ to denote a set of units which are added into $U(S)$ in the $l^{th}$ iteration. Moreover, we have

\[
\left\{ \begin{array}{l}
\sum_{i=1}^{l} a_v(t) < N_{syl} \\
\sum_{i=1}^{l} a_v(t) \geq N_{syl}
\end{array} \right.
\]

Let’s first prove two Lemmas.

Lemma 3.1. In the $l^{th}$ iteration,

\[
Score_{l+1} \geq \frac{a_v(t)}{N_{syl}} (opt - Score(l - 1))
\]

where $opt$ is the value of optimal solution, and

\[
Score_l = \sum_{i=1}^{l} P_l(u)
\]

\[
Score(l) = \sum_{i=1}^{l} Score_i.
\]

Proof. By relaxing the integer constraint on the variables in Eq. (3), the following dual problem can be derived:

\[
g(\Phi_{dual}) = \Gamma N_{syl} + \sum_{u \in U} \theta_u
\]

\[
\lambda_u + \theta_u \geq P_l(u), \quad u \in U
\]

\[
\Gamma \geq \frac{1}{a_v} \sum_{u \in U} \lambda_u c_{uv}, \quad v \in V
\]

where $\Phi_{dual} = \{ \Gamma, \theta_u, \lambda_u, u \in U, v \in V \}$ is the dual variable set for the dual problem.

In the $l^{th}$ iteration of the above greedy procedure where $1 \leq l \leq L$, the dual variable can be defined as

\[
\lambda_u = \begin{cases} 0 & u \in U(S) \\ P_l(u) & u \in U, \ u \notin U(S) \end{cases}
\]

\[
\theta_u = P_l(u) - \lambda_u
\]

\[
\Gamma = \frac{1}{a_v} \sum_{u \in U} \lambda_u c_{uv}
\]

where

\[
v_m = \arg \max_{v \in V} \left( \frac{1}{a_v} \sum_{u \in U} \lambda_u c_{uv} \right).
\]

The above dual variable solution is feasible dual solution in the $l^{th}$ iteration, and in this case

\[
\text{opt} \leq N_{syl} \Gamma + \sum_{u \in U(S)} \theta_u
\]

and

\[
Score_{l+1} \geq \frac{a_v(t)}{N_{syl}} (opt - Score(l - 1)).
\]

Lemma 3.2.

\[
Score(l) \geq \left[ 1 - \prod_{i=1}^{l} \left( 1 - \frac{a_v(i)}{N_{syl}} \right) \right] opt.
\]

Proof. Consider optimal set $S_{opt} = \{ v^{opt}_1, v^{opt}_2, \ldots, v^{opt}_l \}$ and proceed by induction on $l$.

For $l = 1$, $v(1)$ is identified by greedy approach as

\[
v(1) = \arg \max_{v} \frac{Value_v}{a_v}.
\]

It is obvious that

\[
\frac{Value_{v^{opt}_1}}{a_v^{opt}_1} \leq \frac{Value_{v(1)}}{a_v(1)}.
\]

Therefore,

\[
\text{opt} = \sum_{i=1}^{l} Value_{v^{opt}_i} \leq \frac{N_{syl}}{a_v(1)} Value_{v(1)}.
\]

For the $(l + 1)^{th}$ iteration, we have

\[
Score(l + 1) \geq Score(l) + \frac{a_v(t+1)}{N_{syl}} (opt - Score(l)) \geq \left[ 1 - \prod_{i=1}^{l} \left( 1 - \frac{a_v(i)}{N_{syl}} \right) \right] opt.
\]

where the first inequality comes from Lemma 3.1 and the second is due to the induction hypothesis.

Given Lemma 3.1 and Lemma 3.2, our main theorem is as follows:
Theorem 3.3.

\[ \text{Score}(L) \geq \left( 1 - \frac{1}{e^\alpha} \right) \text{opt} \quad (7) \]

where

\[ \sum_{i=1}^L a_{v(i)} \leq N_{\text{syl}} \]
\[ \sum_{i=1}^{L+1} a_{v(i)} > N_{\text{syl}} \]
\[ \alpha = 1 - \max_v a_v \]

\[ \frac{N_{\text{syl}}}{N_{\text{syl}} - \text{max}_v (a_v)} \]

Proof. Because the algebraic mean does not exceed the geometric mean, we have

\[ \prod_{i=1}^L \left( 1 - \frac{a_{v(i)}}{N_{\text{syl}}} \right) \leq \left[ \frac{1}{L} \sum_{i=1}^L \left( 1 - \frac{a_{v(i)}}{N_{\text{syl}}} \right) \right]^L \]
\[ = \left[ 1 - \frac{1}{L} \sum_{i=1}^L a_{v(i)} \right]^L. \]

Since

\[ N_{\text{syl}} - \text{max}_v (a_v) < \sum_{i=1}^L a_{v(i)} \]

then we have

\[ \prod_{i=1}^L \left( 1 - \frac{a_{v(i)}}{N_{\text{syl}}} \right) \leq \left[ 1 - \frac{1}{L} \left( N_{\text{syl}} - \text{max}_v (a_v) \right) \right]^L. \]

Define \( \alpha = \frac{N_{\text{syl}} - \text{max}_v (a_v)}{N_{\text{syl}}} \), then by using Lemma 3.2, we have

\[ \text{Score}(L) \geq \left[ 1 - \frac{1}{e^\alpha} \right] \text{opt} \]
\[ \geq \left[ 1 - \frac{1}{L} \right] \text{opt} \]
\[ \geq \left( 1 - \frac{1}{e^\alpha} \right) \text{opt} . \]

\[ \Box \]

4. Rank-Predicted Pseudo-Greedy Approach to Solving MCKP

Our rank predicted pseudo-greedy approach differs from the above greedy approach mainly in two ways. First, we adopt a pseudo-greedy strategy instead of a greedy one, in which the sentences with the scores more than \( \beta \) times of the maximum are selected. This was inspired by the work in [6]. Second, each sentence \( v \) in the candidate text corpus is ranked by its predicted score, and the sentence with the highest rank is examined first. This was inspired by the work in [12, 2, 10, 11, 4]. The detailed procedure is described in the following.

For each sentence \( v \in V \), compute \( \text{Ratio}_v^0 = \frac{\text{Ratio}_v^0}{\text{Ratio}_{\text{max}}^0} \). Find the maximum and minimum ratio score from \( v \in V \) which are denoted as \( \text{Ratio}_{\text{max}}^0 \) and \( \text{Ratio}_{\text{min}}^0 \) respectively. Let

\[ N = \left\lceil -\log_\beta \frac{\text{Ratio}_{\text{min}}^0}{\text{Ratio}_{\text{max}}^0} + 1 \right\rceil \]

where \( \lceil x \rceil \) is a floor function of \( x \).

Then consider an instance \( \{ V, F \} \) consisting of \( V \) and a family of \( F = \{ S_1, S_2, \ldots, S_N \} \), in which \( S_n \) represents the \( n^{th} \) subset of \( V \) and every sentence of \( V \) belongs to one subset in \( F \); \( V = \cup_{n=1}^N S_n \).

The sentence \( v \in V \) is allocated to \( S_m \) with

\[ m = \left\lceil -\log_\beta \frac{\text{Ratio}_v^0}{\text{Ratio}_{\text{max}}^0} \right\rceil \]

Initialize \( n = 0 \) and repeat the following steps until \( n \geq N \):

- For each sentence \( v \in S_n \) not examined and \( \text{Ratio}_v^0 = \frac{\text{Ratio}_v^0}{\text{Ratio}_{\text{min}}^0} > 0 \),
  - If \( \text{Ratio}_v^0 \geq \beta \times \text{Ratio}_{\text{max}}^0 \), let \( S = S \cup \{ v \} \),
  - Otherwise, let \( S_m = S_m \cup \{ v \} \) with
    \[ m = \left\lceil -\log_\beta \frac{\text{Ratio}_v^0}{\text{Ratio}_{\text{max}}^0} \right\rceil \]
    \[ m < N \].

- Let \( n = n + 1 \) and

\[ \text{Ratio}_{\text{max}}^{n+1} = \beta \times \text{Ratio}_{\text{max}}^n. \quad (9) \]

- Terminate if the total number of syllables in \( S \) is no less than \( N_{\text{syl}} \).

The upper bound of the proposed approach is

\[ \text{Score}(L) \geq \left( 1 - \frac{1}{e^\alpha} \right) \text{opt} \]

which can be proved similarly as Theorem 3.3.

5. Experiments and Results

5.1. Experimental Setup

In order to evaluate the performance improvement of the proposed approach compared with the traditional greedy approach, a series of experiments are conducted. A large-scale text corpus consisting of 1200000 Chinese sentences is used as the back-end text corpus to generate a set of sentences with given number of syllables by resolving problem in Eq. (2). This back-end text corpus is a subset of a Chinese newspaper text corpus. A set of 2366 triphones is used as the coverage unit set. The corresponding occurrence frequencies of coverage units are derived from a
Table 1: Comparison of objective function scores in solving the problem in Eq. (2) by using the proposed rank-predicted pseudo-greedy approach (RPPGA) and the traditional greedy approach.

<table>
<thead>
<tr>
<th>No. of Syllables</th>
<th>400</th>
<th>800</th>
<th>1200</th>
<th>1600</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPPGA</td>
<td>0.848</td>
<td>0.933</td>
<td>0.964</td>
<td>0.978</td>
<td>0.987</td>
</tr>
<tr>
<td>Greedy</td>
<td>0.850</td>
<td>0.935</td>
<td>0.964</td>
<td>0.979</td>
<td>0.988</td>
</tr>
</tbody>
</table>

Table 2: Comparison of user CPU time (in sec) in solving the problem in Eq. (2) by using the proposed rank-predicted pseudo-greedy approach (RPPGA) and the traditional greedy approach.

<table>
<thead>
<tr>
<th>No. of Syllables</th>
<th>400</th>
<th>800</th>
<th>1200</th>
<th>1600</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPPGA</td>
<td>1367</td>
<td>1949</td>
<td>2401</td>
<td>2759</td>
<td>3019</td>
</tr>
<tr>
<td>Greedy</td>
<td>21907</td>
<td>35321</td>
<td>45222</td>
<td>51815</td>
<td>55303</td>
</tr>
</tbody>
</table>

Chinese lexicon consisting of 2756 stock names listed in Stock Exchange in Hong Kong, Shanghai and Shenzhen, respectively. The proposed approach with $\beta = 0.9$ is compared with that of greedy approach using following two performance metrics:

- the objective function score;
- the user CPU time to solve the problem in Eq. (2). The user CPU time is measured by running the experiments on a PC with a 2GHz Intel Pentium-4 CPU (512KB L2 cache) and 512MB main memory.

A comparison of objective function score is summarized in Table 1. Using both approaches, the objective function score is improved as the number of syllables increases and the speed of improvement decreases. Moreover, the similar objective function scores can be achieved by two approaches. A comparison of user CPU time summarized in Table 2 demonstrates the proposed approach is much more efficient than the greedy approach. Furthermore, the increasing speed of the execution time decreases with the increasing number of syllable with reasons as follows:

- for the greedy approach, a lot of candidate texts with $Value_v = 0$ are removed from the back-end text corpus in the earlier iterations and less candidate texts exist for selection in the later iterations;
- for the proposed approach, the value defined as Eq. (9) is an upper bound of $\text{Ratio}_{\text{max}}$, which causes some qualified texts not selected but re-examined. If a small number of syllables is required, the text selection is finished in the early iterations, in which the upper bound is not exact and much execution time are wasted. As the number of syllables increases, such unnecessary re-examinations decrease accordingly due to a more tight upper bound.

6. Summary

In this paper, text selection problem is first formulated as a maximum coverage problem with a Knapsack constraint. An efficient rank-predicted pseudo-greedy approach is then proposed to solve this problem. Experiments on a Chinese text selection task are conducted to verify the efficiency of the proposed approach. Experimental results show that our proposed approach can improve significantly the text selection speed yet without sacrificing the coverage score compared with traditional greedy approach.

7. References