Realtime Event Summarization

Lingting Lin, Yunjie Wang, Chen Lin

March 1, 2018

1 Introduction

2 Related Work

3 Static Summarization

3.1 Problem Definition

Suppose that we have N tweets to be summarized, within which M tweets are credible and relevant. We are going to select a few representative posts from the tweet universe to form the summary. To model this problem, we use a vector $\tilde{\mathbf{x}} \in R^N$, where each element $\tilde{\mathbf{x}}_j \in \{0,1\}$ is a binary variable. If a tweet i is chosen as the abstractive sentence, the corresponding $\tilde{\mathbf{x}}_i = 1$. Otherwise, we set $\tilde{\mathbf{x}}_i = 0$. We use another N-dim vector $\tilde{\mathbf{c}} \in R^M$ to describe the loss of choosing each tweet as a candidate. $\tilde{\mathbf{A}} \in R^{M \times N}$ is a similarity matrix, where $\tilde{a}_{i,j}$ is the similarity between a credible and relevant tweet i and a candidate tweet j in the tweet universe. $\mathbf{b} \in R^M$ is a weight vector, where b_i indicates the importance of i being covered in the summary. Our objective is to

$$\min \tilde{\mathbf{c}}^T \tilde{\mathbf{x}} \text{ subject to } \tilde{\mathbf{A}}\tilde{\mathbf{x}} \geq \mathbf{b}, \tilde{\mathbf{x}} \in \{0,1\}$$

3.2 Methodology Overview

We first transform it to a standard form of bounded linear programming problem by making the following adjustments: $\mathbf{c} = [\tilde{\mathbf{c}}, \mathbf{0}], \mathbf{x} = [\tilde{\mathbf{x}}, \mathbf{z}]^T, \mathbf{A} = [\tilde{\mathbf{A}}, -\mathbf{I}]$, where $\mathbf{I}$ is the $M \times M$ identity matrix. Therefore we have

$$\min \mathbf{cx} \text{ subject to } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \tilde{\mathbf{x}} \leq \mathbf{1} \quad (1)$$

We modify the LP so that there is an easy choice of basic solution. We start by solving

$$\min \mathbf{e^T s} \text{ subject to } \mathbf{Ax} + \mathbf{Is} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \tilde{\mathbf{x}} \leq \mathbf{1}, \mathbf{s} \geq \mathbf{0} \quad (2)$$

where $\mathbf{e}$ is the vector of all ones, $\mathbf{s}$ are called artificial variables,and $b_i \geq \mathbf{0}$. To solve the above LP relaxation, we adopt the bounded simplex method. In each iterate, only M variables are selected as the basic points. Suppose $\mathbf{A}_{,j}$ is the jth column, and the corresponding columns of basic variables in $\mathbf{A}$ are denoted as $\mathbf{A_B}$,others are denoted as $\mathbf{A_N}$. Non-basic variables are either at upper bound ($j \in U$) or at lower bound ($j \in L$).

Define $\mathbf{c}^{'} = [\mathbf{e}, \mathbf{0}], \mathbf{x}^{'} = [\mathbf{x}, \mathbf{s}]^T$ and $\mathbf{A}^{'} = [\mathbf{A}, \mathbf{I}]$ so that the constraints of the modified LP can be written as $\mathbf{A}^{'}\mathbf{x}^{'} = \mathbf{b}, \mathbf{x}^{'} \geq \mathbf{0}, \tilde{\mathbf{x}} \leq \mathbf{1}$.

Step 1: Let $\mathbf{B}$ be the indices of the artificial variables and set the all non-basic variables to be $x_j = 0$. Then $\mathbf{B}$ is a basis,since the corresponding columns of $\mathbf{A}^{'}_{\mathbf{B}}$ are $\mathbf{I}$, the identity,the corresponding basic feasible solution is $\mathbf{x} = \mathbf{0}, \mathbf{z} = \mathbf{b}$.

Step 2: Pricing. $\mathbf{y} = \mathbf{B}^{T^{-1}}\mathbf{e_B}$

Step 3: Compute $\bar{c_j} = c_j - y^T\mathbf{A}^{'}_{,j}$. If $x_j = 0, \bar{c_j} > 0$ and $x_j = 1, \bar{c_j} < 0$, then the problem is solved. Else pick q the most contradictive variable, i.e. the most negative $\bar{c_q}$ for $x_q = 0$ as the entering index(step 4 in Algorithm 1).

Step 4: Choose the outing index(step 5 to step 27 in Algorithm 1) and update the basis and the value of variables.

Step 5: If all artificial variables are non-basic or some artificial variables are in the basis but all $x_{z_i} = 0$ then go to Step 6,otherwise return to Step 2.

Step 6: If some artificial variables are in the basis,remove them from basis. Change $\mathbf{c}^{'} = [\tilde{\mathbf{c}}, \mathbf{0}]$, repeat Step 2 to Step 4.

4 Update Summarization

$$\min c_0x_0 + c_1x_1 s.t. \begin{bmatrix} A & D \\ D^T & B \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \geq \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \tag{3}$$

$D^Tx_0 \geq b_1$

Input: $\mathbf{c} = [\mathbf{e}, \mathbf{0}], \mathbf{x}' = [\mathbf{x}, \mathbf{s}]^T$ and $\mathbf{A}' = [\mathbf{A}, \mathbf{I}]$

Output: $\mathbf{x}'_{\mathbf{B}} = {\mathbf{A}'_{\mathbf{B}}}^{-1}(\mathbf{b} - \mathbf{u})$, where $\mathbf{u} = \Sigma_{j \in U} \mathbf{A}'_{,j}$

Pricing $\mathbf{y} = {\mathbf{B}^T}^{-1} \mathbf{e}_{\mathbf{B}}$;

Compute $\bar{c_j} = c_j - y^T \mathbf{A}'_{,j}$;

while $\exists x_j = 0, \bar{c_j} < 0$ *or* $x_j = 1, \bar{c_j} < 0$ **do**

  $q = \arg\max_j \{\bar{c_j} \forall j \in \{\mathbf{z_N}, \mathbf{s_N}, \mathbf{x_L}\}, -\bar{c_j} \forall j \in \mathbf{x_U}\}$;

  $d = \mathbf{B}^{-1} \mathbf{A}'_{,q}$;

  if $q \in \{\mathbf{z_N}, \mathbf{s_N}\}$ **then**

    $x_q^{new} = \min_i \left\{ \{\frac{x_{\mathbf{B}_i}}{d_i}, \frac{s_{\mathbf{B}_i}}{d_i}, \frac{z_{\mathbf{B}_i}}{d_i}\} \forall d_i > 0, \{\frac{x_{\mathbf{B}_i} - 1}{d_i}\} \forall d_i < 0 \right\}$;

    $\mathbf{x}'^{new}_{\mathbf{B}^{old}} = \mathbf{x}'^{old}_{\mathbf{B}^{old}} - d x_q^{new}$;

    $p = \arg\min_i$;

    $\mathbf{B}^{new} \leftarrow \mathbf{B}^{old} - \{p\} \bigcup \{q\}$;

  end

  if $q \in L$ **then**

    $x_q^{new} = \min_i \left\{ \{\frac{x_{\mathbf{B}_i}}{d_i}, \frac{s_{\mathbf{B}_i}}{d_i}, \frac{z_{\mathbf{B}_i}}{d_i}\} \forall d_i > 0, \{\frac{x_{\mathbf{B}_i} - 1}{d_i}\} \forall d_i < 0, 1 \right\}$;

    $\mathbf{x}'^{new}_{\mathbf{B}^{old}} = \mathbf{x}'^{old}_{\mathbf{B}^{old}} - d x_q^{new}$;

    if $x_q^{new} \neq 1$ **then**

      $p = \arg\min_i$;

      $\mathbf{B}^{new} \leftarrow \mathbf{B}^{old} - \{p\} \bigcup \{q\}$;

    end

  end

  if $q \in U$ **then**

    $x_q^{new} = 1 - \min_i \left\{ \{\frac{x_{\mathbf{B}_i}}{-d_i}, \frac{s_{\mathbf{B}_i}}{-d_i}, \frac{z_{\mathbf{B}_i}}{-d_i}\} \forall d_i < 0, \{\frac{1 - x_{\mathbf{B}_i}}{d_i}\} \forall d_i > 0, 1 \right\}$;

    $\mathbf{x}'^{new}_{\mathbf{B}^{old}} = \mathbf{x}'^{old}_{\mathbf{B}^{old}} + d(1 - x_q^{new})$;

    if $x_q^{new} \neq 0$ **then**

      $p = \arg\min_i$;

      $\mathbf{B}^{new} \leftarrow \mathbf{B}^{old} - \{p\} \bigcup \{q\}$;

    end

  end

end

Algorithm 1: the bounded simplex method