# Text Summarization while Maximizing Multiple Objectives with Lagrangian Relaxation

Masaaki Nishino, Norihito Yasuda, Tsutomu Hirao,
Jun Suzuki, and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation
{nishino.masaaki,yasuda.n,hirao.tsutomu,suzuki.jun,
nagata.masaaki}@lab.ntt.co.jp

**Abstract.** We show an extractive text summarization method that solves an optimization problem involving the maximization of multiple objectives. Though we can obtain high quality summaries if we solve the problem exactly with our formulation, it is NP-hard and cannot scale to support large problem size. Our solution is an efficient and high quality approximation method based on Lagrangian relaxation (LR) techniques. In experiments on the DUC'04 dataset, our LR based method matches the performance of state-of-the-art methods.

## 1 Introduction

We introduce an extractive text summarization method that formulates the task as a combinatorial optimization problem. Previous work on this line can be divided into two groups: the combination of maximization of relevance and minimization of redundancy (e.g. [5]), and graph-based methods (e.g. [6]). We set up an objective for the optimization problem that combines these two approaches to improve summary quality. The problem is NP-hard if it is solved exactly as an integer linear programming (ILP) problem, and we cannot obtain the solution in feasible time if the problem size is large. We thus propose a fast and high quality optimization heuristic based on the use of *Lagrangian Relaxation (LR)*. In experiments on the DUC'04 dataset, our method demonstrates performance comparable to that of the state-of-the-art non NP-hard methods.

## 2 Objective Functions and ILP Formulations

We use $D = \{s_1, \ldots, s_N\}$ to represent the set of documents, where $s_i$ is the $i$-th sentence in $D$. $L_{\max}$ is the limit length of the summary. We use $N$-dimensional binary vectors $x, y, z \in \{0,1\}^N$ to represent the summary. If the $i$-th sentence $s_i$ is in the summary represented by $x$, then $x_i = 1$, otherwise $x_i = 0$. We formalize the text summarization problem as the optimization problem of maximizing an objective function that measures summary quality. Our objective function combines the three objectives of *relevance*, *redundancy*, and *coverage*.

P. Serdyukov et al. (Eds.): ECIR 2013, LNCS 7814, pp. 772–775, 2013.

*Relevance.* Relevance of a summary is the amount of relevant information the summary contains. We use $f(x)$ as the relevance score of summary $x$, and define it as the sum of the relevance scores of the sentences contained in $x$. We set $w_i$ as the relevance score of the $i$-th sentence $s_i$, and define the objective as $f(x) = \sum_{i=1}^{N} w_i x_i$. We use $w_i = \frac{1}{pos(s_i)} \sum_{t \in s_i} \log(\chi^2(t) + 1)$, where $pos(s_i)$ is the position of sentence $s_i$ in the document that contains it, and $\chi^2(t)$ is the chi-square value of word $t$. chi-square value is used for extracting relevance sentences in [3]. The problem of maximizing $f(x)$ under the summary length constraint can be formulated as a knapsack problem, and the exact solution can be found efficiently by applying dynamic programming (DP).

*Redundancy.* We measure the redundancy of a summary by the number of different bigrams contained in the summary, since a summary with less unique bigrams contains duplicated contents in it. We use $g(x)$ as the relevance score of summary $x$, and we set $g(x) = \sum_{b_i \in \Gamma(x)} b_i u_i$, where $u_i \in \{0, 1\}$ is a binary variable and $u_i = 1$ if the $i$-th bigram is contained in summary $x$. $\Gamma(x)$ is the set of all unique bigrams contained in the summary, and $b_i$ is the weight of bigram $u_i$, which is defined as the number of documents containing the $i$-th bigram.

*Coverage.* For measuring coverage, we use the score function used in [6]; they introduced similarity between every pair of sentences contained in a document set, and then evaluated a summary by how well the sentences contained in the summary cover the sentences in the document set. We use $h(x)$ as the coverage score of summary $x$, and define it as $h(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} e_{ij} v_{ij}$, where $v_{ij}$ is a binary variable and $v_{ij} = 1$ if the $i$-th sentence is contained in summary $x$ and $j$-th sentence is not, and $j$-th sentence is regarded to be "covered" by the $i$-th sentence. Here we assume two constraints; a sentence can cover other sentences if it is contained in the summary, i.e., $\sum_j v_{ij} \geq 0$ if $x_i = 1$ otherwise $\sum_j v_{ij} = 0$ for all $i$, and every sentence must not be covered by more than two sentences, i.e. $\sum_i v_{ij} \leq 1$ for all $j$. $e_{ij}$ is the score for the $j$-th sentence covered by the $i$-th sentence. We define score $e_{ij}$ as the asymmetric score defined in [6].

*ILP Formulation.* Combining the above three objectives, we formulate text summarization as a combinatorial optimization problem that can be formulated in Integer Linear Programming terms as Fig.1, which maximizes the objective with regard to $x$. $c_i$ is the length of the $i$-th sentence, $M$ is the number of different bigrams in $D$, and $a_{ij}$ is a binary constant that equals 1 if the $i$-th sentence

$$\begin{aligned}
\text{maximize} \;\; & f(x) + g(x) + h(x) = \sum_{i=1}^{N} w_i x_i + \sum_{i=1}^{M} b_i u_i + \sum_{i=1}^{N} \sum_{j=1}^{N} e_{ij} v_{ij} \\
\text{subject to} \;\; & (1) \sum_{i=1}^{N} c_i x_i \leq L_{\max}, \quad (2) \forall j : \sum_{i=1}^{N} a_{ij} x_i \geq u_j, \quad (3) \forall j : \sum_{i=1}^{N} v_{ij} \leq 1 \\
& (4) \forall i, j : x_i \geq z_{ij}, \quad (5) \forall i : z_{ii} = x_i, \quad (6) \forall i, j : x_i, u_i, v_{ij} \in \{0, 1\}
\end{aligned}$$

**Fig. 1.** ILP formulation

contains the $j$-th bigram, otherwise $a_{ij} = 0$. Constraint (1) ensures that the length of a generated summary is less than the length limit $L_{\max}$. Constraint (2) addresses bigram occurrence and states that $u_i$ can take the value of 1 only if at least one sentence that contains the corresponding bigram is in the summary. Constraints (3) to (5) address the covering relation between sentences. Constraint (6) ensures variables are binary. Though we can obtain the exact solution with ILP solvers, the problem is NP-hard.

## 3 Lagrangian Relaxation

We introduce LR-based heuristics for obtaining a good approximate solution in much shorter time than is possible with ILP. If we can maximize the three previously noted objectives, $f(x)$, $g(y)$, and $h(z)$, for independent $N$ dimensional vectors $x$, $y$, and $z$, the problem becomes easier. However, we actually need to consider the maximization problem under the constraint of $x = y = z$, which is difficult. Lagrangian relaxation overcomes this problem by easing the constraint on the equality of variables; we first define the following Lagrangian:

$$L(\lambda, \mu, x, y, z) = f(x) + g(y) + h(z) + \textstyle\sum_{i=1}^{N} (\lambda_i(x_i - y_i) + \mu_i(x_i - z_i)),$$

where $\lambda_i, \mu_i$ $(1 \le i \le N)$ are Lagrange multipliers. Solving the original problem by minimizing the Lagrangian dual:

$$\begin{aligned} L(\lambda, \mu) &= \max_{x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}} L(\lambda, \mu, x, y, z) \\ &= \max_{x \in \mathcal{X}} \{f(x) + \textstyle\sum_{i=1}^{N} (\lambda_i + \mu_i) x_i\} + \max_{y \in \mathcal{Y}} \{g(y) - \textstyle\sum_{i=1}^{N} \lambda_i y_i\} + \max_{z \in \mathcal{Z}} \{h(z) - \textstyle\sum_{i=1}^{N} \mu_i z_i\}, \end{aligned}$$

where we use $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$ to represent the sets of all possible summaries that satisfy the corresponding constraints in Fig.1. The problem of minimizing $L(\lambda, \mu)$ can be solved efficiently by using the subgradient method. We show the flow of our algorithm in Alg. 1. $K$ is the maximum number of iterations. We solve the problem by repeatedly maximizing the three objectives independently (Line 3) and updating the Lagrange multipliers (Line 6). These maximization processes can be done efficiently by applying DP for $f(x)$, and greedy search method for $g(y)$ and $h(z)$. Since we design $g(y)$ and $h(z)$ to be *submodular functions*, we can find good approximate solutions with a simple greedy algorithm. We terminate the process if the three solutions of the subproblems are the same (Line 4) or the $K$-th iteration is reached (Line 7).

## 4 Evaluation and Conclusion

We use the DUC'04 dataset for evaluating our approach. We set the summary length limit to 665 bytes, the setting used in [4]. We evaluated the summary using ROUGE-(1/2) recall(R) and F-measure(F) with ROUGE version 1.5.5 [2]. We compared the results of the LR-based method with three state-of-the-art methods that are not NP-hard; submodular function maximization (Lin'11) [4] and McDonald's DP-like method (McDonald'07) [5], and CLASSY [1]. As a reference,

**Algorithm 1.** The Lagrangian-relaxation-based text summarization algorithm.

1: $\lambda_i^{(0)} \leftarrow 0, \mu_i^{(0)} \leftarrow 0$ **for** $i = 1$ to $N$
2: **for** $k = 1$ to $K$ **do**
3: $x^{(k)} \leftarrow \underset{x \in \mathcal{X}}{\operatorname{argmax}}\{f(x) + \sum_{i=1}^{N} (\lambda_i^{(k-1)} + \mu_i^{(k-1)})x_i\}$
$y^{(k)} \leftarrow \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\{g(y) - \sum_{i=1}^{N} \lambda_i^{(k-1)} y_i\}$, $z^{(k)} \leftarrow \underset{z \in \mathcal{Z}}{\operatorname{argmax}}\{h(z) - \sum_{i=1}^{N} \mu_i^{(k-1)} z_i\}$
4: **if** $x_i^{(k)} = y_i^{(k)} = z_i^{(k)}$ **for all** $i \in \{1, \ldots, N\}$ **then return** $x^{(k)}$
5: **for** $i = 1$ to $N$ **do**
6: $\lambda_i^{(k)} \leftarrow \lambda_i^{(k-1)} - \delta_k(x_i^{(k)} - y_i^{(k)})$, $\mu_i^{(k)} \leftarrow \mu_i^{(k-1)} - \delta_k(x_i^{(k)} - z_i^{(k)})$
7: **return** $x^{(K)}$

**Table 1.** ROUGE scores, objectives, and computation time

| Method | ROUGE-1 | | ROUGE-2 | | Objective | Time(sec.) |
|---|---|---|---|---|---|---|
| | F | R | F | R | | |
| Proposed (LR) | 0.390 | 0.397 | 0.098 | 0.096 | 14.0 | 3.46 |
| Lin'11 | 0.389 | 0.394 | – | – | – | – |
| CLASSY | 0.377 | 0.382 | 0.092 | 0.091 | – | – |
| Mcdonald'07 | 0.362 | 0.338 | 0.081 | 0.086 | – | – |
| Proposed (ILP) | 0.396 | 0.401 | 0.099 | 0.101 | 14.8 | 109 |

we also solving the ILP problem using ILOG CPLEX. We implemented the LR method in C++. We set $K = 500$, and $\delta^{(k)} = 0.5/k$, where $k$ is the number of iterations. We show the results in Tab.1. We can see Proposed(LR) offers comparable ROUGE-1 (F/R) scores to Lin'11. Lin'11 needs K-means clustering beforehand and its performance is sensitive to the settings of the initial values, while the proposed method avoids this issue. We can see that LR offers a high objective value that is about 95% of the exact solution obtained by ILP, while the average computational time is much shorter. The difference of the computational time tends to be large with larger document sets.

We see our ILP formulation and LR-based method both match the performance of the state-of-the-art methods on experiments on the DUC'04 dataset.

## References


1. Conroy, J., Schlesinger, J., Goldstein, J., Oleary, D.: Left-brain/right-brain multi-document summarization. In: Proceedings of DUC 2004 (2004)
2. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Proceedings of Workshop on Text Summarization Branches Out (2004)
3. Lin, C.Y., Hovy, E.: The automated acquisition of topic signatures for text summarization. In: Proceedings of the 18th COLING (2000)
4. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th ACL/HLT (2011)
5. McDonald, R.: A Study of Global Inference Algorithms in Multi-document Summarization. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 557–564. Springer, Heidelberg (2007)
6. Takamura, H., Okumura, M.: Text summarization model based on the budgeted median problem. In: Proceedings of the 18th CIKM (2009)