Multi-document Summarization via Budgeted Maximization of Submodular Functions

Hui Lin
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
hlin@ee.washington.edu

Jeff Bilmes
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
bilmes@ee.washington.edu

Abstract

We treat the text summarization problem as maximizing a submodular function under a budget constraint. We show, both theoretically and empirically, a modified greedy algorithm can efficiently solve the budgeted submodular maximization problem near-optimally, and we derive new approximation bounds in doing so. Experiments on DUC'04 task show that our approach is superior to the best-performing method from the DUC'04 evaluation on ROUGE-1 scores.

1 Introduction

Automatically generating summaries from large text corpora has long been studied in both information retrieval and natural language processing. There are several types of text summarization tasks. For example, if an input query is given, the generated summary can be query-specific, and otherwise it is generic. Also, the number of documents to be summarized can vary from one to many. The constituent sentences of a summary, moreover, might be formed in a variety of different ways — summarization can be conducted using either *extraction* or *abstraction*, the former selects only sentences from the original document set, whereas the latter involves natural language generation. In this paper, we address the problem of generic extractive summaries from clusters of related documents, commonly known as *multi-document summarization*.

In extractive text summarization, textual units (e.g., sentences) from a document set are extracted to form a summary, where grammaticality is assured at the local level. Finding the optimal summary can be viewed as a combinatorial optimization problem which is NP-hard to solve (McDonald, 2007). One of the standard methods for this problem is called *Maximum Marginal Relevance* (MMR) (Dang, 2005)(Carbonell and Goldstein, 1998), where a greedy algorithm selects the most relevant sentences, and at the same time avoids redundancy by removing sentences that are too similar to already selected ones. One major problem of MMR is that it is non-optimal because the decision is made based on the scores at the current iteration. McDonald (2007) proposed to replace the greedy search of MMR with a globally optimal formulation, where the basic MMR framework can be expressed as a knapsack packing problem, and an integer linear program (ILP) solver can be used to maximize the resulting objective function. ILP Algorithms, however, can sometimes either be expensive for large scale problems or themselves might only be heuristic without associated theoretical approximation guarantees.

In this paper, we study graph-based approaches for multi-document summarization. Indeed, several graph-based methods have been proposed for extractive summarization in the past. Erkan and Radev (2004) introduced a stochastic graph-based method, *LexRank*, for computing the relative importance of textual units for multi-document summarization. In LexRank the importance of sentences is computed based on the concept of eigenvector centrality in the graph representation of sentences. Mihalcea and Tarau also proposed an eigenvector centrality algorithm on weighted graphs for document summarization (Mihalcea and Tarau, 2004). Mihalcea et al. later applied Google's *PageRank* (Brin and Page, 1998) to natural language processing tasks ranging from automatic keyphrase extraction and word sense disambiguation, to extractive summarization (Mihalcea et al., 2004; Mihalcea, 2004). Recent work in (Lin et al., 2009) presents a graph-based approach where an undirected weighted graph is built for the document to be summarized, and vertices represent the candidate sentences and edge weights represent the similarity between sentences. The summary extraction procedure is done by maximizing a submodular set function under a cardinality constraint.

Inspired by (Lin et al., 2009), we perform summarization by maximizing submodular functions under a *budget* constraint. A budget constraint is natural in summarization task as the length of the summary is often restricted. The length (byte budget) limitation represents the real world scenario where summaries are displayed using only limited computer screen real estate. In practice, the candidate textual/linguistic units might not have identical costs (e.g., sentence lengths vary). Since a cardinality constraint is a special case (a budget constraint with unity costs), our approach is more general than (Lin et al., 2009). Moreover, we propose a modified greedy algorithm (Section 4) and both theoretically (Section 4.1) and empirically (Section 5.1) show that the algorithm solves the problem near-optimally, thanks to submodularity. Regarding summarization performance, experiments on DUC'04 task show that our approach is superior to the best-performing method in DUC'04 evaluation on ROUGE-1 scores (Section 5).

2 Background on Submodularity

Consider a set function $f : 2^V \rightarrow \mathbb{R}$, which maps subsets $S \subseteq V$ of a finite ground set V to real numbers. $f(\cdot)$ is called normalized if $f(\emptyset) = 0$, and is *monotone* if $f(S) \leq f(T)$ whenever $S \subseteq T$. $f(\cdot)$ is called *submodular* (Lovasz, 1983) if for any $S, T \subseteq V$, we have

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T). \quad (1)$$

An equivalent definition of submodularity is the property of *diminishing returns*, well-known in the field of economics. That is, $f(\cdot)$ is submodular if for any $R \subseteq S \subseteq V$ and $s \in V \setminus S$,

$$f(S \cup \{s\}) - f(S) \leq f(R \cup \{s\}) - f(R). \quad (2)$$

Eqn. 2 states that the "value" of s never increases in the contexts of ever larger sets, exactly the property of diminishing returns. This phenomenon arises naturally in many other contexts as well. For example, the Shannon entropy function is submodular in the set of random variables. Submodularity, moreover, is a discrete analog of convexity (Lovasz, 1983). As convexity makes continuous functions more amenable to optimization, submodularity plays an essential role in combinatorial optimization.

Many combinatorial optimization problems can be solved optimally or near-optimally in polynomial time only when the underlying function is submodular. It has been shown that any submodular function can be *minimized* in polynomial time (Schrijver, 2000)(Iwata et al., 2001). Maximization of submodular functions, however, is an NP-complete optimization problem but fortunately, some submodular maximization problems can be solved near-optimally. A famous result is that the maximization of a monotone submodular function under a cardinality constraint can be solved using a greedy algorithm (Nemhauser et al., 1978) within a constant factor (0.63) of being optimal. A constant-factor approximation algorithm has also been obtained for maximizing monotone submodular function with a knapsack constraint (see Section 4.2). Feige et.al. (2007) studied unconstrained maximization of a arbitrary submodular functions (not necessarily monotone). Kawahara et.al. (2009) proposed a cutting-plane method for optimally maximizing a submodular set function under a cardinality constraint, and Lee et.al. (2009) studied non-monotone submodular maximization under matroid and knapsack constraints.

3 Problem Setup

In this paper, we study the problem of maximizing a submodular function under budget constraint, stated formally below:

$$\max_{S \subseteq V} \left\{ f(S) : \sum_{i \in S} c_i \leq \mathcal{B} \right\} \quad (3)$$

where V is the ground set of all linguistic units (e.g., sentences) in the document, S is the extracted summary (a subset of V), c_i is the non-negative cost of selecting unit i and $\mathcal{B}$ is our budget, and submodular function $f(\cdot)$ scores the summary quality.

The budgeted constraint arises naturally since often the summary must be length limited as mentioned above. In particular, the budget $\mathcal{B}$ could be the maximum number of words allowed in any summary, or alternatively the maximum number of bytes of any summary, where c_i would then be either number of words or the number of bytes in sentence i.

To benefit from submodular optimization, the objective function measuring the summary quality must be submodular. In general, there are two ways to apply submodular optimization to any application domain. One way is to force submodularity on an application, leading to an artificial and poorly performing objective function even if it can be optimized well. The alternative is to address applications where submodularity naturally applies. We are fortunate in that, like convexity in the continuous domain, submodularity seems to arise naturally in a variety of discrete domains, and as we will see below, extractive summarization is one of them. As mentioned in Section 1, our approach is graph-based, not only because a graph is a natural representation of the relationships and interactions between textual units, but also because many submodular functions are well defined on a graph and can naturally be used in measuring the summary quality.

Suppose certain pairs (i, j) with $i, j \in V$ are similar and the similarity of i and j is measured by a non-negative value $w_{i,j}$. We can represent the entire document with a weighted graph (V, E), with non-negative weights $w_{i,j}$ associated with each edge $e_{i,j}, e \in E$. One well-known graph-based submodular function that measures the similarity of S to the remainder $V \setminus S$ is the graph-cut function:

$$f_{\text{cut}}(S) = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j}. \qquad (4)$$

In multi-document summarization, redundancy is a particularly important issue since textual units from different documents might convey the same information. A high quality (small and meaningful) summary should not only be informative about the remainder but also be compact (non-redundant). Typically, this goal is expressed as a combination of maximizing the information coverage and minimizing the redundancy (as used in MMR (Carbonell and Goldstein, 1998)). Inspired by this, we use the following objective by combining a λ-weighted penalty term with the graph cut function:

$$f_{\text{MMR}}(S) = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j} - \lambda \sum_{i,j \in S : i \neq j} w_{i,j}, \lambda \geq 0.$$

Luckily, this function is still submodular as both the graph cut function and the redundancy term are submodular. Neither objective, however, is monotone, something we address in Theorem 3. Although similar to the MMR objective function, our approach is different since 1) ours is graph-based and 2) we formalize the problem as submodular function maximization under the budget constraint where a simple greedy algorithm can solve the problem guaranteed near-optimally.

4 Algorithms

Algorithm 1 Modified greedy algorithm

```
$G \leftarrow \emptyset$
$U \leftarrow V$
while $U \neq \emptyset$ do
    $k \leftarrow \arg\max_{\ell \in U} \frac{f(G \cup \{\ell\}) - f(G)}{(c_\ell)^r}$
    $G \leftarrow G \cup \{k\}$ if $\sum_{i \in G} c_i + c_k \leq B$ and $f(G \cup \{k\}) - f(G) \geq 0$
    $U \leftarrow U \setminus \{k\}$
end while
$v^* \leftarrow \arg\max_{v \in V, c_v \leq \mathcal{B}} f(\{v\})$
return $G_f = \arg\max_{S \in \{\{v^*\}, G\}} f(S)$
```

Inspired by (Khuller et al., 1999), we propose Algorithm 1 to solve Eqn. (3). The algorithm sequentially finds unit k with the largest ratio of objective function gain to scaled cost, i.e., $(f(G \cup \{\ell\}) - f(G))/c_\ell^r$, where $r > 0$ is the scaling factor. If adding k increases the objective function value while not violating the budget constraint, it is then selected and otherwise bypassed. After the sequential selection, set G is compared to the within-budget singleton with the largest objective value, and the larger of the two becomes the final output.

The essential aspect of a greedy algorithm is the design of the greedy heuristic. As discussed in (Khuller et al., 1999), a heuristic that greedily selects the k that maximizes $(f(G \cup \{k\}) - f(G))/c_k$ has an unbounded approximation factor. For example, let $V = \{a, b\}$, $f(\{a\}) = 1, f(\{b\}) = p$,

$c_a = 1, c_b = p+1$, and $\mathcal{B} = p+1$. The solution obtained by the greedy heuristic is $\{a\}$ with objective function value 1, while the true optimal objective function value is p. The approximation factor for this example is then p and therefore unbounded.

We address this issue by the following two modifications to the naive greedy algorithms. The first one is the final step (line 8 and 9) in Algorithm 1 where set G and singletons are compared. This step ensures that we could obtain a constant approximation factor for $r = 1$ (see the proof in the Appendix).

The second modification is that we introduce a scaling factor r to adjust the scale of the cost. Suppose, in the above example, we scale the cost as $c_a = 1^r, c_b = (p+1)^r$, then selecting a or b depends also on the scale r, and we might get the optimal solution using a appropriate r. Indeed, the objective function values and the costs might be uncalibrated since they might measure different units. E.g., it is hard to say if selecting a sentence of 15 words with an objective function gain of 2 is better than selecting sentence of 10 words with gain of 1. Scaling can potentially alleviate this mismatch (i.e., we can adjust r on development set). Interestingly, our theoretical analysis of the performance guarantee of the algorithm also gives us guidance about how to scale the cost for a particular problem (see Section 4.1).

4.1 Analysis of performance guarantee

Although Algorithm 1 is essentially a simple greedy strategy, we show that it solves Eqn. (3) globally and near-optimally, by exploiting the structure of submodularity. As far as we know, this is a new result for submodular optimization, not previously stated or published before.

Theorem 1. *For normalized monotone submodular function $f(\cdot)$, Algorithm 1 with $r = 1$ has a constant approximation factor as follows:*

$$f(G_f) \geq \left(1 - e^{-\frac{1}{2}}\right) f(S^*), \tag{5}$$

where S^* is an optimal solution.

Proof. See Appendix. □

Note that an α-approximation algorithm for an optimization problem is a polynomial-time algorithm that for *all* instances of the problem produces a solution whose value is within a factor of α of the value of the an optimal solution. So Theorem 1 basically states that the solution found by Algorithm 1 can be at least as good as $(1 - 1/\sqrt{e}) f(S^*) \approx 0.39 f(S^*)$ even in the worst case. A constant approximation bound is good since it is true for all instances of the problem, and we always know how good the algorithm is guaranteed to be without any extra computation. For $r \neq 1$, we resort to instance-dependent bound where the approximation can be easily computed per problem instance.

Theorem 2. *With normalized monotone submodular $f(\cdot)$, for $i = 1, \ldots, |G|$, let v_i be the ith unit added into G and G_i is the set after adding v_i. When $0 \leq r \leq 1$,*

$$f(G_i) \geq \left(1 - \prod_{k=1}^{i} \left(1 - \frac{c_{v_k}^r}{\mathcal{B}^r |S^*|^{1-r}}\right)\right) f(S^*) \tag{6}$$

$$\geq \left(1 - \prod_{k=1}^{i} \left(1 - \frac{c_{v_k}^r}{\mathcal{B}^r |V|^{1-r}}\right)\right) f(S^*) \tag{7}$$

and when $r \geq 1$,

$$f(G_i) \geq \left(1 - \prod_{k=1}^{i} \left(1 - \left(\frac{c_{v_k}}{\mathcal{B}}\right)^r\right)\right) f(S^*). \tag{8}$$

Proof. See Appendix. □

Theorem 2 gives bounds for a specific instance of the problem. Eqn. (6) requires the size $|S^*|$, which is unknown, requiring us to estimate an upper bound of the cardinality of the optimal set S^*. Obviously, $|S^*| \leq |V|$, giving us Eqn. (7). A tighter upper bound is obtained, however, by sorting the costs. That is, let $c_{[1]}, c_{[2]}, \ldots, c_{[|V|]}$ be the sorted sequence of costs in nondecreasing order, giving $|S^*| < m$ where $\sum_{k=1}^{m-1} c_{[i]} \leq \mathcal{B}$ and $\sum_{k=1}^{m} c_{[i]} > \mathcal{B}$. In this case, the computation cost for the bound estimation is $O(|V| \log |V|)$, which is quite feasible.

Note that both Theorem 1 and 2 are for monotone submodular functions while our practical objective function, i.e. f_{MMR}, is not guaranteed everywhere monotone. However, our theoretical results still holds for f_{MMR} with high probability in practice. Intuitively, in summarization tasks, the summary is usually small compared to the ground set size ($|S| \ll |V|$). When $|S|$ is small, f_{MMR} is monotone and our theoretical results still hold. Precisely, assume that all edge weights are bounded: $w_{i,j} \in [0, 1]$ (which is the case for cosine similarity between non-negative vectors). Also assume that edges weights are independently identically distributed with mean μ, i.e. $\mathbb{E}(w_{i,j}) = \mu$. Given a budget $\mathcal{B}$, assume the maximum possible size of a solution is K. Let $\alpha = 2\lambda + 1$, and $\beta = 2K - 1$. Notice that $\beta \ll |V|$ for our summarization task. We have the following theorem:

Theorem 3. *Algorithm 1 solves the summarization problem near-optimally (i.e. Theorem 1 and Theorem 2 hold) with high probability of at least*

$$1 - \exp\left\{-\frac{2(|V| - (\alpha+1)\beta)^2\mu^2}{|V| + (\alpha^2 - 1)\beta} + \ln K\right\}$$

Proof. Omitted due to space limitation. □

4.2 Related work

Algorithms for maximizing submodular function under budget constraint (Eqn. (3)) have been studied before. Krause (2005) generalized the work by Khuller et al.(1999) on budgeted maximum cover problem to the submodular framework, and showed a $\frac{1}{2}(1 - 1/e)$-approximation algorithm. The algorithm in (Krause and Guestrin, 2005) and (Khuller et al., 1999) is actually a special case of Algorithm 1 when $r = 1$, and Theorem 1 gives a better bound (i.e., $(1 - 1/\sqrt{e}) > \frac{1}{2}(1 - 1/e)$) in this case. There is also a greedy algorithm with partial enumerations (Sviridenko, 2004; Krause and Guestrin, 2005) factor $(1 - 1/e)$. This algorithm, however, is too computationally expensive and thus not practical for real world applications (the computation cost is $O(|V|^5)$ in general). When each unit has identical cost, the budget constraint reduces to cardinality constraint where a greedy algorithm is known to be a $(1-1/e)$-approximation algorithm (Nemhauser et al., 1978) which is the best that can be achieved in polynomial time (Feige, 1998) if P $\neq$ NP. Recent work (Takamura and Okumura, 2009) applied the maximum coverage problem to text summarization (without apparently being aware that their objective is submodular) and studied a similar algorithm to ours when $r = 1$ and for the non-penalized graph-cut function. This problem, however, is a special case of constrained submodular function maximization.

5 Experiments

We evaluated our approach on the data set of DUC'04 (2004) with the setting of task 2, which is a multi-document summarization task on English news articles. In this task, 50 document clusters are given, each of which consists of 10 documents. For each document cluster, a short multi-document summary is to be generated. The summary should not be longer than 665 bytes including spaces and punctuation, as required in the DUC'04 evaluation. We used DUC'03 as our development set. All documents were segmented into sentences using a script distributed by DUC. ROUGE version 1.5.5 (Lin, 2004), which is widely used in the study of summarization, was used to evaluate summarization performance in our experiments [1]. We focus on ROUGE-1 (unigram) F-measure scores since it has demonstrated strong correlation with human annotation (Lin, 2004).

The basic textual/linguistic units we consider in our experiments are sentences. For each document cluster, sentences in all the documents of this cluster forms the ground set V. We built semantic graphs for each document cluster based on cosine similarity, where cosine similarity is computed based on the TF-IDF (term frequency, inverse document frequency) vectors for the words in the sentences. The cosine similarity measures the similarity between sentences, i.e., $w_{i,j}$.

Here the IDF values were calculated using all the document clusters. The weighted graph was built by connecting vertices (corresponding to sentences) with weight $w_{i,j} > 0$. Any unconnected vertex was removed from the graph, which is equivalent to pre-excluding certain sentences from the summary.

5.1 Comparison with exact solution

In this section, we empirically show that Algorithm 1 works near-optimally in practice. To determine how much accuracy is lost due to approximations, we compared our approximation algorithms with an exact solution. The exact solutions were obtained by Integer Linear Programming (ILP). Solving arbitrary ILP is an NP-hard problem. If the size of the problem is not too large, we can sometimes find the exact solution within a manageable time

[1] Options used: -a -c 95 -b 665 -m -n 4 -w 1.2

using a branch-and-bound method. In our experiments, MOSEK was used as our ILP solver.

We formalize Eqn. (3) as an ILP by introducing indicator (binary) variables $x_{i,j}, y_{i,j}, i \neq j$ and z_i for $i, j \in V$. In particular, $z_i = 1$ indicates that unit i is selected, i.e., $i \in S$, $x_{i,j} = 1$ indicates that $i \in S$ but $j \notin S$, and $y_{i,j} = 1$ indicates both i and j are selected. Adding constraints to ensure a valid solution, we have the following ILP formulation for Eqn. (3) with objective function $f_{\text{MMR}}(S)$:

$$\max \sum_{i \neq j, i,j \in V} w_{i,j} x_{i,j} - \lambda \sum_{i \neq j, i,j \in V} w_{i,j} y_{i,j}$$

$$\text{subject to: } \sum_{i \in V} c_i z_i \leq \mathcal{B},$$

$$x_{i,j} - z_i \leq 0, x_{i,j} + z_j \leq 1, z_i - z_j - x_{i,j} \leq 0,$$

$$y_{i,j} - z_i \leq 0, y_{i,j} - z_j \leq 0, z_i + z_j - y_{i,j} \leq 1,$$

$$x_{i,j}, y_{i,j}, z_i \in \{0, 1\}, \forall i \neq j, i, j \in V$$

Note that the number of variables in the ILP formulation is $O(|V|^2)$. For a document cluster with hundreds of candidate textual units, the scale of the problem easily grows involving tens of thousands of variables, making the problem very expensive to solve. For instance, solving the ILP exactly on a document cluster with 182 sentences (as used in Figure 1) took about 17 hours while our Algorithm 1 finished in less than 0.01 seconds.

We tested both approximate and exact algorithms on DUC'03 data where 60 document clusters were used (30 TDT document clusters and 30 TREC document clusters), each of which contains 10 documents on average. The true approximation factor was computed by dividing the objective function value found by Algorithm 1 over the optimal objective function value (found by ILP). The average approximation factors over the 58 document clusters (ILP on 2 of the 60 document clusters failed to finish) are shown in Table 1, along with other statistics. On average Algorithm 1 finds a solution that is over 90% as good as the optimal solution for many different r values, which backs up our claim that the modified greedy algorithm solves the problem near-optimally, even occasionally optimally (Figure 1 shows one such example).

The higher objective function value does not always indicate higher ROUGE-1 score. Indeed, rather than directly optimizing ROUGE, we optimize a surrogate submodular function that indicates the quality of a summary. Optimality in the submodular function does not necessary indicate optimality in ROUGE score. Nevertheless, we will show that our approach outperforms several other approaches in terms of ROUGE. We note that ROUGE is itself a surrogate for true human-judged summary quality, it might possibly be that f_{MMR} is a still better surrogate — we do not consider this possibility further in this work, however.

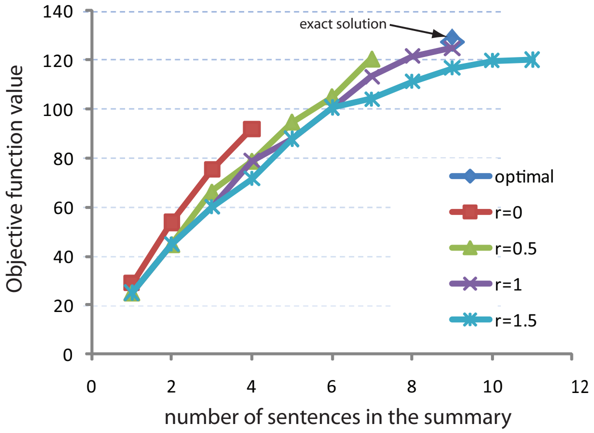

Figure 1: Application of Algorithm 1 when summarizing document cluster d30001t in the DUC'04 dataset with summary size limited to 665 bytes. The objective function was f_{MMR} with $\lambda = 2$. The plots show the achieved objective function as the number of selected sentences grows. The plots stop when in each case adding more sentences violates the budget. Algorithm 1 with $r = 1$ found the optimal solution exactly.

5.2 Summarization Results

We used DUC'03 (as above) for our development set to investigate how r and λ relate to the ROUGE-1 score. From Figure 2, the best performance is achieved with $r = 0.3, \lambda = 4$. Using these settings, we applied our approach to the DUC'04 task. The results, along with the results of other approaches, are shown in Table 2. All the results in Table 2 are presented as ROUGE-1 F-measure scores. [2]

We compared our approach to two other well-

[2] When the evaluation was done in 2004, ROUGE was still in revision 1.2.1, so we re-evaluated the DUC'04 submissions using ROUGE v1.5.5 and the numbers are slightly different from the those reported officially.

Table 1: Comparison of Algorithm 1 to exact algorithms on DUC'03 dataset. All the numbers shown in the table are the average statistics (mean/std). The "true" approximation factor is the ratio of objective function value found by Algorithm 1 over the ILP-derived true-optimal objective value, and the approximation bounds were estimated using Theorem 2.

	Approx. factor		ROUGE-1
	true	bound	(%)
exact	1.00	-	33.60/5.05
$r = 0.0$	0.65/0.15	$\geq$0.19/0.08	33.50/5.94
$r = 0.1$	0.71/0.15	$\geq$0.24/0.08	33.68/6.03
$r = 0.3$	0.88/0.11	$\geq$0.37/0.06	34.77/5.49
$r = 0.5$	0.96/0.04	$\geq$0.48/0.05	34.33/5.94
$r = 0.7$	0.98/0.02	$\geq$0.56/0.05	34.08/5.41
$r = 1.0$	0.98/0.02	$\geq$0.65/0.04	33.32/5.14
$r = 1.2$	0.97/0.02	$\geq$0.48/0.05	32.54/4.69

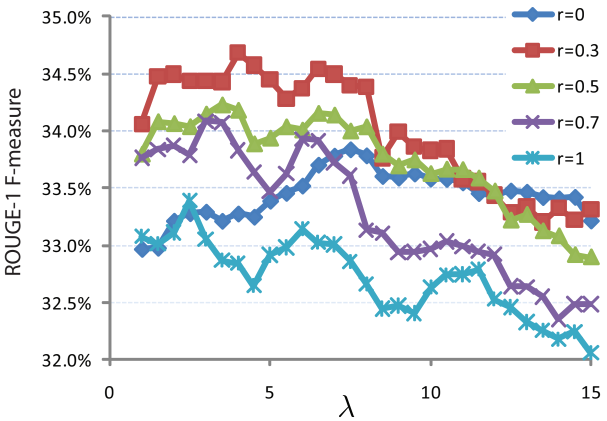

Figure 2: Different combinations of r and λ for f_{MMR} related to ROUGE-1 score on DUC'03 task 1.

known graph-based approaches, LexRank and PageRank. LexRank was one of the participating system in DUC'04, with peer code 104. For PageRank, we implemented the recursive graph-based ranking algorithm ourselves. The importance of sentences was estimated in an iterative way as in (Brin and Page, 1998)(Mihalcea et al., 2004). Sentences were then selected based on their importance rankings until the budget constraint was violated. The graphs used for PageRank were *exactly* the graphs in our submodular approaches (i.e., an undirected graph). In both cases, submodular summarization achieves better ROUGE-1 scores. The improvement is statistically significant by the Wilcoxon signed rank test at level $p < 0.05$. Our approach also outperforms the best system (Conroy et al., 2004), peer code 65 in the DUC'04 evaluation although not as significant ($p < 0.08$). The reason might be that DUC'03 is a poor representation of DUC'04 — indeed, by varying r and λ over the ranges $0 \leq r \leq 0.2$ and $5 \leq \lambda \leq 9$ respectively, the DUC'04 ROUGE-1 scores were all $> 38.8\%$ with the best DUC'04 score being 39.3%.

Table 2: ROUGE-1 F-measure results (%)

Method	ROUGE-1 score
peer65 (best system in DUC04)	37.94
peer104 (LexRank)	37.12
PageRank	35.37
Submodular ($r = 0.3, \lambda = 4$)	**38.39**

6 Appendix

We analyze the performance guarantee of Algorithm 1. We use the following notation: S^* is the optimal solution; G_f is the final solution obtained by Algorithm 1; G is the solution obtained by the greedy heuristic (line 1 to 7 in Algorithm 1); v_i is the ith unit added to G, $i = 1, \ldots, |G|$; G_i is the set obtained by greedy algorithm after adding v_i (i.e., $G_i = \cup_{k=1}^{i}\{v_k\}$, for $i = 1, \ldots, |G|$, with $G_0 = \emptyset$ and $G_{|G|} = G$); $f(\cdot) : 2^V \rightarrow \mathbb{R}$ is a monotone submodular function; and $\rho_k(S)$ is the gain of adding k to S, i.e., $f(S \cup \{k\}) - f(S)$.

Lemma 1. *$\forall X, Y \subseteq V$,*

$$f(X) \leq f(Y) + \sum_{k \in X \setminus Y} \rho_k(Y). \tag{9}$$

Proof. See (Nemhauser et al., 1978) □

Lemma 2. *For $i = 1, \ldots, |G|$, when $0 \leq r \leq 1$,*

$$f(S^*) - f(G_{i-1}) \leq \frac{\mathcal{B}^r |S^*|^{1-r}}{c_{v_i}^r}(f(G_i) - f(G_{i-1})), \tag{10}$$

and when $r \geq 1$,

$$f(S^*) - f(G_{i-1}) \leq \left(\frac{\mathcal{B}}{c_{v_i}}\right)^r (f(G_i) - f(G_{i-1})) \tag{11}$$

Proof. Based on line 4 of Algorithm 1, we have

$$\forall u \in S^* \setminus G_{i-1}, \frac{\rho_u(G_{i-1})}{c_u^r} \leq \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r}.$$

Thus when $0 \le r \le 1$,

$$\begin{aligned}
\sum_{u\in S^*\setminus G_{i-1}} \rho_u(G_{i-1}) &\le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} \sum_{u\in S^*\setminus G_{i-1}} c_u^r \\
&\le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} |S^* \setminus G_{i-1}| \left(\frac{\sum_{u\in S^*\setminus G_{i-1}} c_u}{|S^* \setminus G_{i-1}|}\right)^r \\
&\le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} |S^*|^{1-r} \left(\sum_{u\in S^*\setminus G_{i-1}} c_u\right)^r \\
&\le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} |S^*|^{1-r} \mathcal{B}^r,
\end{aligned}$$

where the second inequality is due to the concavity of $g(x) = x^r, x > 0, 0 \le r \le 1$. The last inequality uses the fact that $\sum_{u\in S^*} c_u \le \mathcal{B}$. Similarly, when $r \ge 1$,

$$\begin{aligned}
\sum_{u\in S^*\setminus G_{i-1}} \rho_u(G_{i-1}) \le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} &\sum_{u\in S^*\setminus G_{i-1}} c_u^r \\
\le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} &\left(\sum_{u\in S^*\setminus G_{i-1}} c_u\right)^r \\
\le \frac{\rho_{v_i}(G_{i-1})}{c_{v_i}^r} &\mathcal{B}^r.
\end{aligned}$$

Applying Lemma 1, i.e., let $X = S^*$ and $Y = G_{i-1}$, the lemma immediately follows. □

The following is a proof of Theorem 2.

Proof. Obviously, the theorem is true when $i = 1$ by applying Lemma 2.

Assume that the theorem is true for $i-1, 2 \le i \le |G|$, we show that it also holds for i. When $0 \le r \le 1$,

$$\begin{aligned}
&f(G_i) = f(G_{i-1}) + (f(G_i) - f(G_{i-1})) \\
&\ge f(G_{i-1}) + \frac{c_{v_i}^r}{\mathcal{B}^r|S^*|^{1-r}}(f(S^*) - f(G_{i-1})) \\
&= \left(1 - \frac{c_{v_i}^r}{\mathcal{B}^r|S^*|^{1-r}}\right) f(G_{i-1}) + \frac{c_{v_i}^r}{\mathcal{B}^r|S^*|^{1-r}} f(S^*) \\
&\ge \left(1 - \frac{c_{v_i}^r}{\mathcal{B}^r|S^*|^{1-r}}\right)\left(1 - \prod_{k=1}^{i-1}\left(1 - \frac{c_{v_k}^r}{\mathcal{B}^r|S^*|^{1-r}}\right)\right) \\
&f(S^*) + \frac{c_{v_i}^r}{\mathcal{B}^r|S^*|^{1-r}} f(S^*) \\
&= \left(1 - \prod_{k=1}^{i}\left(1 - \frac{c_{v_k}^r}{\mathcal{B}^r|S^*|^{1-r}}\right)\right) f(S^*).
\end{aligned}$$

The case when $r \ge 1$ can be proven similarly. □

Now we are ready to prove Theorem 1.

Proof. Consider the following two cases:

Case 1: $\exists v \in V$ such that $f(\{v\}) > \frac{1}{2}f(S^*)$. Then it is guaranteed that $f(G_f) \ge f(\{v\})) > \frac{1}{2}f(S^*)$ due line 9 of Algorithm 1.

Case 2: $\forall v \in V$, we have $f(\{v\}) \le \frac{1}{2}f(S^*)$. We consider the following two sub-cases, namely Case 2.1 and Case 2.2:

Case 2.1: If $\sum_{v\in G} c_v \le \frac{1}{2}\mathcal{B}$, then we know that $\forall v \notin G, c_v > \frac{1}{2}\mathcal{B}$ since otherwise we can add a $v \notin G$ into G to increase the objective function value without violating the budget constraint. This implies that there is at most one unit in $S^* \setminus G$ since otherwise we will have $\sum_{v\in S^*} c_v > \mathcal{B}$. By assumption, we have $f(S^* \setminus G) \le \frac{1}{2}f(S^*)$. Submodularity of $f(\cdot)$ gives us:

$$f(S^* \setminus G) + f(S^* \cap G) \ge f(S^*),$$

which implies $f(S^* \cap G) \ge \frac{1}{2}f(S^*)$. Thus we have

$$f(G_f) \ge f(G) \ge f(S^* \cap G) \ge \frac{1}{2}f(S^*),$$

where the second inequality follows from monotonicity.

Case 2.2: If $\sum_{v\in G} c_v > \frac{1}{2}\mathcal{B}$, for $0 \le r \le 1$, using Theorem 2, we have

$$\begin{aligned}
f(G) &\ge \left(1 - \prod_{k=1}^{|G|}\left(1 - \frac{c_{v_k}^r}{\mathcal{B}^r|S^*|^{1-r}}\right)\right) f(S^*) \\
&\ge \left(1 - \prod_{k=1}^{|G|}\left(1 - \frac{c_{v_k}^r|S^*|^{r-1}}{2^r\left(\sum_{k=1}^{|G|} c_{v_k}\right)^r}\right)\right) f(S^*) \\
&\ge \left(1 - \left(1 - \frac{|S^*|^{r-1}}{2^r|G|^r}\right)^{|G|}\right) f(S^*) \\
&\ge \left(1 - e^{-\frac{1}{2}\left(\frac{|S^*|}{2|G|}\right)^{r-1}}\right) f(S^*)
\end{aligned}$$

where the third inequality uses the fact (provable using Lagrange multipliers) that for $a_1, \ldots, a_n \in \mathbb{R}^+$ such that $\sum_{i=1}^n a_i = \alpha$, function

$$1 - \prod_{i=1}^{n}\left(1 - \frac{\beta a_i^r}{\alpha^r}\right)$$

achieves its minimum of $1 - (1 - \beta/n^r)^n$ when $a_1 = \cdots = a_n = \alpha/n$ for $\alpha, \beta > 0$. The last inequality follows from $e^{-x} \ge 1 - x$.

In all cases, we have

$$f(G_f) \ge \min\left\{\frac{1}{2}, 1 - e^{-\frac{1}{2}\left(\frac{|S^*|}{2|G|}\right)^{r-1}}\right\} f(S^*)$$

In particular, when $r = 1$, we obtain the constant approximation factor, i.e.

$$f(G_f) \ge \left(1 - e^{-\frac{1}{2}}\right) f(S^*)$$

□

Acknowledgments

This work is supported by an ONR MURI grant (No. N000140510388), the Companions project (IST programme under EC grant IST-FP6-034434), and the National Science Foundation under grant IIS-0535100. We also wish to thank the anonymous reviewers for their comments.

References

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR*.

J.M. Conroy, J.D. Schlesinger, J. Goldstein, and D.P. O'leary. 2004. Left-brain/right-brain multi-document summarization. In *Proceedings of the Document Understanding Conference (DUC 2004)*.

H.T. Dang. 2005. Overview of DUC 2005. In *Proceedings of the Document Understanding Conference*.

2004. Document understanding conferences (DUC). http://www-nlpir.nist.gov/projects/duc/index.html.

G. Erkan and D.R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

U. Feige, V. Mirrokni, and J. Vondrak. 2007. Maximizing non-monotone submodular functions. In *Proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.

U. Feige. 1998. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652.

G. Goel, , C. Karande, P. Tripathi, and L. Wang. 2009. Approximability of Combinatorial Problems with Multi-agent Submodular Cost Functions. FOCS.

S. Iwata, L. Fleischer, and S. Fujishige. 2001. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777.

Yoshinobu Kawahara, Kiyohito Nagano, Koji Tsuda, and Jeff Bilmes. 2009. Submodularity cuts and applications. In *Neural Information Processing Society (NIPS)*, Vancouver, Canada, December.

S. Khuller, A. Moss, and J. Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.

A. Krause and C. Guestrin. 2005. A note on the budgeted maximization of submodular functions. *Technical Rep. No. CMU-CALD-05*, 103.

J. Lee, V.S. Mirrokni, V. Nagarajan, and M. Sviridenko. 2009. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing*, pages 323–332. ACM New York, NY, USA.

Hui Lin, Jeff Bilmes, and Shasha Xie. 2009. Graph-based submodular selection for extractive summarization. In *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU)*, Merano, Italy, December.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

L. Lovasz. 1983. Submodular functions and convexity. *Mathematical programming-The state of the art,(eds. A. Bachem, M. Grotschel and B. Korte) Springer*, pages 235–257.

R. McDonald. 2007. A study of global inference algorithms in multi-document summarization. *Lecture Notes in Computer Science*, 4425:557.

R. Mihalcea and P. Tarau. 2004. TextRank: bringing order into texts. In *Proceedings of EMNLP*, Barcelona, Spain.

R. Mihalcea, P. Tarau, and E. Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*.

R. Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 (companion volume)*.

2006. Mosek.

G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14(1):265–294.

A. Schrijver. 2000. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355.

M. Sviridenko. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43.

H. Takamura and M. Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 781–789. Association for Computational Linguistics.