

Graph-based Submodular Selection for Extractive Summarization

Hui Lin¹, Jeff Bilmes¹, Shasha Xie²

¹*Department of Electrical Engineering, University of Washington
Seattle, WA 98195, United States*

{hlin,bilmes}@ee.washington.edu

²*Department of Computer Science, The University of Texas at Dallas,
Richardson, TX 75080, United States*

shasha@hlt.utdallas.edu

Abstract—We propose a novel approach for unsupervised extractive summarization. Our approach builds a semantic graph for the document to be summarized. Summary extraction is then formulated as optimizing submodular functions defined on the semantic graph. The optimization is theoretically guaranteed to be near-optimal under the framework of submodularity. Extensive experiments on the ICSI meeting summarization task on both human transcripts and automatic speech recognition (ASR) outputs show that the graph-based submodular selection approach consistently outperforms the maximum marginal relevance (MMR) approach, a concept-based approach using integer linear programming (ILP), and a recursive graph-based ranking algorithm using Google’s PageRank.

I. INTRODUCTION

Extractive summarization selects salient sentences from original documents and presents them as a summary. Finding the optimal summary can be viewed as a combinatorial optimization problem which is NP-hard to solve [1]. One of the standard methods for this problem is called Maximum Marginal Relevance (MMR) [2][3], 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. In [1], McDonald 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. However, ILP is itself an approximation to the original combinatorial problem and often a theoretical approximation guarantee of any form does not exist.

In this paper, we introduce a graph-based submodular selection approach with theoretical constant-factor approximation guarantees. An undirected weighted graph is built for the document to be summarized. In the graph, 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 defined on the graph under the constraint that only a certain number of sentences can be selected (Sec. III). The maximization is guaranteed to be near-optimal — i.e., the greedy algorithm can find a solu-

tion that is no worse than a constant (~ 0.632) fraction of the optimal value, which is the best we can do in polynomial time unless $P = NP$ (Sec. II). When optimizing the worst-case over multiple submodular functions, we moreover utilize the recently developed SATURATE algorithm [4] which provides theoretical guarantees in this case. Extensive experiments on the ICSI meeting summarization task (Sec. V) show that our new approach outperforms MMR, a concept-based approach using integer linear programming (ILP) [5], and a recursive graph-based ranking algorithm using Google’s PageRank [6].

II. BACKGROUND

A. Submodularity

Consider a set function $f : 2^V \rightarrow \mathbb{R}$, which maps subsets $S \subseteq V$ of a finite set V to real numbers. $f(\cdot)$ is called *submodular* [7] if for any $S, T \subseteq V$,

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

In the task of extractive summarization, V is the set of all sentences in the document, S is the extracted summary (a subset of V), and the function $f(\cdot)$ scores the quality of the summary.

Submodularity is the discrete analog of convexity [7]. As convexity makes continuous functions more amenable to optimization, submodularity plays an essential role in combinatorial optimization. Submodular functions appear in diverse settings including graph cut [8], set covering [9], facility location [10], game theory [11] and sensor placement [4] problems.

An equivalent condition for submodularity is the property of diminishing returns. That is for any $R \subseteq S \subseteq V$ and $s \in V$,

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

Intuitively, this means that the “value” of adding element s to a set never increases as the set gets larger. The canonical example is an urn with colored balls, where the value of the urn is the number of different colors.

B. Submodular Selection

In the framework of submodularity, many combinatorial optimization problems can be solved optimally or near-optimally in polynomial time. It has been shown that any submodular function can be *minimized* in polynomial time [12][13], a property which has been exploited recently in the machine learning community [14][15]. Maximization of submodular functions, however, is an NP-complete optimization problem. Fortunately, maximization of a monotone submodular function under a cardinality constraint can be solved near optimally using a greedy algorithm, which motivates our submodular selection approach herein.

In particular, we are interested in the following constrained maximization problem: we want to select a good subset S of the whole set V that maximizes some objective function, given the constraint that the size of S is no larger than K (our budget). Such selection problems arise in many applications. For instance, in active learning, we wish to acquire labels only for the most informative subset of the unlabeled data (which is usually abundant) given limited budget and/or time for labeling. In extractive summarization, the problem is to find a subset S (sentences) that is most representative of the whole set V (document), given the constraint that $|S| \leq K$, i.e. only a small number of sentences can be used in the summary, or the summary should achieve a certain word compression rate.

We formally state the selection problem as follows:

$$\max_{S \subseteq V} \{f(S) : |S| \leq K\}. \quad (3)$$

While NP hard, this problem can be approximately solved when $f(\cdot)$ is submodular using a simple greedy forward-selection algorithm. The algorithm starts with $S = \emptyset$, and iteratively adds the element $s^* \in V \setminus S$ that yields the greatest increment of the objective function value:

$$s^* \in \operatorname{argmax}_{s \in V \setminus S} f(S \cup \{s\}) - f(S). \quad (4)$$

This repeats until either $|S| = K$ or no further increment occurs.

When $f(\cdot)$ is non-decreasing, normalized, and submodular, this simple greedy algorithm performs near-optimally, i.e.:

Theorem 1. Nemhauser et al. 1978 [16]. *If submodular function $f(\cdot)$ satisfies: i) non-decreasing: for all $S_1 \subseteq S_2 \subseteq V$, $f(S_1) \leq f(S_2)$; ii) normalized: $f(\emptyset) = 0$, then the set S_G^* obtained by the greedy algorithm is no worse than a constant fraction $(1 - 1/e)$ away from the optimal value:*

$$f(S_G^*) \geq (1 - 1/e) \max_{S \subseteq V: |S| \leq K} f(S). \quad (5)$$

The greedy algorithm, moreover, is likely to be the best we can do in polynomial time, unless $P = NP$.

Theorem 2. Feige 1998 [9] *Unless $P=NP$, there is no polynomial-time algorithm that guarantees a solution S^* with*

$$f(S^*) \geq (1 - 1/e + \epsilon) \max_{S \subseteq V: |S| \leq K} f(S), \epsilon > 0. \quad (6)$$

III. GRAPH-BASED SUBMODULAR SELECTION

As mentioned above, extractive summarization can be cast as a subset selection problem. Suppose we have a set of sentences $V = \{1, 2, \dots, N\}$, where certain sentences pairs (i, j) are similar and the similarity of i and j is measured by a non-negative value $w_{i,j}$. We can represent the whole document using a weighted graph $G = (V, E)$, with non-negative weights $w_{i,j}$ associated with each edge (i, j) . Extractive summarization then finds a subset S that best represents the entire set V . To leverage submodularity, we introduce several submodular set functions, each of which measures how “representative” S is of the entire set V .

A. Common Submodular Set Functions

Two well-known submodular functions can be used to measure the representativeness of S to the entire set V . The first one is the uncapacitated facility location function [10]:

$$f_{\text{facility}}(S) = \sum_{i \in V} \max_{j \in S} w_{i,j}. \quad (7)$$

This measures the similarity of S to the whole set V . We can also measure the similarity of S to the remainder, i.e., the graph cut function:

$$f_{\text{cut}}(S) = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j}. \quad (8)$$

B. Worst-case Objective function

For $i \in V$, the function

$$g_i(S) = \max_{j \in S} w_{i,j} \quad (9)$$

measures the similarity of sentence i to the selected (summary) set S . Actually, $f_{\text{facility}}(S) = \sum_i g_i(S)$ can be viewed as an average of the similarities of *all* the sentences in the document to the summary. In some cases, i.e., when the weights of the graph are noisy, optimizing the average may be inadequate (as we see in our experiments). As an extreme example, consider a document where all the sentences are highly related to each other, except only one of them is about a somewhat different but important topic. Obviously, the ideal summary for this document should also contain this sentence. Optimizing the average, however, is unlikely to include this sentence in the final summary. This can be resolved if we optimize the worst rather than the average case, motivating our next objective function where we maximize the similarity of the least similar sentence to the summary:

$$f_{\text{worst}}(S) = \min_{i \in V} g_i(S) = \min_{i \in V} \max_{j \in S} w_{i,j}. \quad (10)$$

Note that $g_i(S)$ is submodular for all i (see Sec. VI). However, f_{worst} is not necessarily submodular. Fortunately, recent development in robust submodular selection [4] enables us to approximately solve the maximization of f_{worst} with strong theoretical approximation guarantees. See the algorithm section (Sec. III-D) for details.

C. Penalty of redundancy

A high quality summary should not only be informative but also compact. Typically, this goal is expressed as a combination of maximizing the information coverage and minimizing the redundancy. Here, we propose the following objective by combining a penalty term $-\sum_{i \in S} \sum_{j \in S, j \neq i} w_{i,j}$ with the graph cut function:

$$f_{\text{penalty}} = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j} - \lambda \sum_{i,j \in S: i \neq j} w_{i,j}, \lambda \geq 0. \quad (11)$$

This function is still submodular (see Sec. VI). In MMR [3], a similar approach is used, where the algorithm greedily selects sentences that are most similar to the remainder of the document but least similar to the already selected sentences. Our method differs from MMR in that it is graph-based and under a submodular framework. As shown in Sec. V, our approach consistently and significantly outperforms MMR.

D. Algorithms

All the objective functions except $f_{\text{worst}}(\cdot)$ are normalized submodular set functions. In order to benefit from Theorem 1, the objective function should also be nondecreasing. Obviously, the facility location objective function is nondecreasing since $w_{i,j} \geq 0$. For the graph cut objective, the increment of adding k into S is

$$f_{\text{cut}}(S \cup \{k\}) - f_{\text{cut}}(S) = \sum_{i \in V \setminus S} w_{i,k} - \sum_{j \in S \cup \{k\}} w_{k,j},$$

which is not always nonnegative. Fortunately, the proof of Theorem 1 does not use the monotone property for all possible sets [16][17, page 58]. f_{cut} can also respect the conditions for Theorem 1 if $|S| \ll |V|$, which is usually the case in summarization where the extracted summary is usually much smaller than the entire document. Similarly, f_{penalty} can also be non-decreasing in the early stage of selection if λ is not too large (in our experiments, the working values of λ always lie in $[0, 2]$).

Hence for f_{facility} , f_{cut} and f_{penalty} , we use greedy algorithms to solve the extractive summarization problem efficiently and near-optimally. The greedy algorithm for f_{facility} is described in Algorithm 1, where $\rho_i = \max_{j \in S} w_{i,j}$ is used to speed up the algorithm. The algorithms for f_{cut} and f_{penalty} are similar to Algorithm 1 and are thus omitted.

f_{worst} , although not submodular, is a minimization over a set of monotone submodular functions and can be optimized using the SATURATE algorithm introduced in [4]. SATURATE is an efficient algorithm for the robust submodular observation selection problem which guarantees solutions to be at least as informative as the optimal solution, at only a slightly higher cost. Basically, the algorithm maintains an upper bound for the problem as well as a lower bound for a relaxed version of the original problem. It successively improves the upper and lower bounds using a binary search procedure. We give the SATURATE algorithm for f_{worst} in Algorithm 2 and refer readers to [4] for additional details and its theoretical guarantees.

Algorithm 1 Greedy algorithm for f_{facility}

- 1: **Input:** $G = (V, E)$ with weights $w_{i,j}$ on edge (i, j) ; K : the number of sentences to be selected
 - 2: **Initialization:** $S = \emptyset$, $\rho_i = 0$, $i = 1, \dots, N$ where $N = |V|$
 - 3: **while** $|S| \leq K$ **do**
 - 4: $k^* = \arg \max_{k \in V \setminus S} \sum_{i \in V, (i,k) \in E} (\max\{\rho_i, w_{i,k}\} - \rho_i)$
 - 5: $S = S \cup \{k^*\}$
 - 6: **for all** $i \in V$ **do**
 - 7: $\rho_i = \max\{\rho_i, w_{i,k^*}\}$
 - 8: **end for**
 - 9: **end while**
-

Algorithm 2 SATURATE algorithm for f_{worst}

- 1: **Input:** $G = (V, E)$ with weights $w_{i,j}$ on edge (i, j) ; K : the number of sentences to be selected
 - 2: **Initialization:** $c_{\min} = 0$, $c_{\max} = \min_{i \in V} \max_{i \in V} w_{i,j}$, $S^* = \emptyset$, $N = |V|$, $\alpha = 1.1$
 - 3: **while** $c_{\max} - c_{\min} > \frac{1}{N}$ **do**
 - 4: $c = (c_{\max} + c_{\min})/2$; $S = \emptyset$
 - 5: Define $\bar{f}_c(S) = \frac{1}{N} \sum_{i \in V} \min\{\max_{j \in S} w_{i,j}, c\}$
 - 6: **while** $\bar{f}_c(S) < c$ **do**
 - 7: $S = S \cup \{\arg \max_{k \in V \setminus S} \bar{f}_c(S \cup \{k\}) - \bar{f}_c(S)\}$
 - 8: **end while**
 - 9: **if** $|S| > \alpha K$ **then**
 - 10: $c_{\max} = c$
 - 11: **else**
 - 12: $c_{\min} = c$, $S^* = S$
 - 13: **end if**
 - 14: **end while**
-

IV. RELATED WORK

Several graph-based methods have been proposed for extractive summarization previously. Erkan and Radev [18] 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 [19]. Mihalcea et al. later applied Google's *PageRank* [20] to natural language processing tasks ranging from automatic keyphrase extraction and word sense disambiguation, to extractive summarization [6][21]. Graph-based ranking algorithms, such as *PageRank*, can be applied to natural language processing applications by building lexical or semantic graphs extracted from the documents to be processed. In [21], *PageRank* was adopted to incorporate edge weights, and the power $P(i)$ (importance) of a sentence i was iteratively computed as

$$P(i) = (1-d) + d \times \sum_{j \in \text{Parent}(i)} w_{j,i} \frac{P(j)}{\sum_{k \in \text{Child}(j)} w_{k,j}}. \quad (12)$$

where d is a parameter usually set between 0 and 1. A summary is then extracted based on the ranking of the sentences.

We implemented and compared the PageRank-based algorithm to our approach in the experiments (Sec. V).

Submodularity has already been successfully used in machine learning as well as in speech and language processing. Narasimhan and Bilmes have shown how submodularity can help structure learning for graphical models [14] and clustering of words in language models [15]. Krause et al. explored robust submodular observation selection for sensor placement [17]. Our recent work [22] also introduced a submodular framework for active learning in automatic speech recognition.

V. EXPERIMENTS

A. Experimental setup

We evaluated our approach on the ICSI meeting corpus [23]. There are 75 meeting recordings in this corpus. Each meeting is about one hour long and has multiple speakers. Since we focus on unsupervised meeting summarization, only the development set and the test set were used in the evaluation, both of which consist of 6 meetings as used in [24] and [25].

Both human transcripts and automatic speech recognition (ASR) outputs are available for this corpus, where the ASR output is obtained from the state-of-the-art SRI conversational telephone speech system [26], having an overall word error rate of about 38.2%. Three reference summaries from different annotators for each meeting were used for the test set, while for the development set, only one reference summary was used. The lengths of the reference summaries are not fixed and vary across annotators and meetings. The average word compression ratio of the reference summaries is 14.3% with a mean deviation 2.9%. In our experiments, all methods were evaluated when extracting summaries with word compression ratios varying from 13% to 17%.

ROUGE [27], which is widely used in the study of speech summarization [24][28][29], was used to evaluate summarization performance in our experiments. To be consistent with previous work, we provide ROUGE-1 (unigram) F-measure results for all experiments.

B. Semantic graphs

We built semantic graphs for each meeting recording in the development and test sets, on both human transcripts and ASR outputs. Two methods were used.

The first method is based on cosine similarity, where the cosine similarity between two sentences D_i and D_j is:

$$w_{i,j} = \text{sim}(D_i, D_j) = \frac{\sum_k t_{ik}t_{jk}}{\sqrt{\sum_k t_{ik}^2} \times \sqrt{\sum_k t_{jk}^2}}, \quad (13)$$

where t_{ik} is the TF-IDF (term frequency, inverse document frequency) weight for word W_k in sentence D_i . Here the IDF values were calculated using transcripts of 75 meetings. For both human transcripts and ASR outputs, we split each of the 75 meetings into multiple topics based on manual topic segment annotations, and then used these new “documents” to calculate the IDF values. 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. On average, about 99% of the sentences are preserved in the graph for all meeting recordings on both human transcripts and ASR outputs.

The second method we used is based on the ROUGE score itself. Words with low TF-IDF weights (stop words) were initially removed from the sentences, and then the ROUGE-1 F-measure scores between each stop-word-removed sentences were used as the similarity measure. The removal of stop words resulted in many “empty” sentences. These empty sentences always had zero connections in the graph, and thus were pre-excluded from the final summary, yielding sparse final graphs. On human transcripts, the graph preserves only 45% of the sentences while 40% are preserved in the graph for ASR outputs.

C. Comparison to other approaches

We compared our approach to MMR [3] and a global optimization framework using integer linear programming (referred as “ILP” in the rest of this paper) recently proposed in [5]. The setups of both MMR and ILP approaches were the same as the baseline systems introduced in [25].

In addition to non-graph-based approaches, we also compared our method to the recursive graph-based ranking algorithm using PageRank. The importance of sentences was estimated in an iterative way using Equation 12, where the value for d was set at 0.85 as in [20][6]. Iteration stopped when the relative difference from the successive iteration fell below 0.01%. As introduced in [6], the graph in this algorithm can be represented as: (a) an undirected graph where a vertex’s parents and children are both those vertices connected to it (PageRank-U); (b) a directed weighted graph with the orientation of edges set from sentence to sentences that follow in the text (PageRank-F); or (c) a directed weighted graph with edges oriented from a sentence to previous sentences in the text (PageRank-B).

D. Results and discussion

Results using the cosine similarity graph are shown in Table I for human transcripts and in Table II for ASR outputs. Results with graphs built on ROUGE scores are illustrated in Table III and Table IV for human transcripts and ASR outputs, respectively. All the results are presented as ROUGE-1 F-measure scores under different word compression ratios ranging from 13% to 17%. In each table, results on both the development (dev.) set and test set of all methods are shown. There are 4 categories of methods: MMR, ILP, PageRank and submodular selection. A number is bold if it beats the results of *all* the methods in the other three categories under the same word compression rate. The best result under the same word compression rate is marked with a “*”.

As we can see, for all tables, all the bold and starred numbers appear in the rows for submodular selection, indicating that our graph-based submodular selection outperforms MMR, ILP and PageRank consistently in all cases.

TABLE I
ROUGE-1 F-MEASURE RESULTS (%) FOR DIFFERENT WORD
COMPRESSION RATIOS FOR HUMAN TRANSCRIPTS (REF) ON BOTH DEV.
SET AND TEST SET WITH GRAPHS BASED ON COSINE SIMILARITY.

REF.G-cosine.DEV.	ROUGE-1 F-Measure (%)				
	13%	14%	15%	16%	17%
Word comp. ratio					
MMR	66.28	66.81	67.06	66.90	66.64
ILP	66.46	67.20	67.98	68.30	67.82
PageRank-U	49.54	49.84	50.10	50.20	50.17
PageRank-F	61.41	61.93	62.12	62.15	61.80
PageRank-B	63.01	63.71	64.36	64.21	64.40
Submodular- $f_{facility}$	66.71	67.11	68.03	67.92	67.74
Submodular- f_{cut}	60.36	61.45	61.89	62.39	62.57
Submodular- f_{worst}	69.02*	69.29*	69.42*	69.24*	68.50*
Submodular- $f_{penalty}$	66.70	67.26	67.00	66.73	66.34
REF.G-cosine.TEST	ROUGE-1 F-Measure (%)				
Word comp. ratio	13%	14%	15%	16%	17%
MMR	64.67	65.69	66.23	66.69	66.70
ILP	66.11	67.08	67.84	68.35	68.82
PageRank-U	51.90	52.90	53.41	53.49	53.56
PageRank-F	60.86	61.50	62.19	62.41	62.37
PageRank-B	63.15	63.89	64.50	64.93	64.83
Submodular- $f_{facility}$	66.06	66.99	67.31	67.62	67.46
Submodular- f_{cut}	60.95	62.17	63.11	63.62	63.91
Submodular- f_{worst}	67.89*	68.57*	69.14*	69.23*	69.01*
Submodular- $f_{penalty}$	67.45	67.89	68.30	68.35	67.97

Note that optimizing f_{cut} performs poorly on the graph built on human transcripts using cosine similarity (Table I). One reason is that this graph is quite noisy. Meeting summarization is particularly challenging due to the presence of disfluencies. In the human transcripts, disfluencies are precisely transcribed where filler words (such as “so”, “yeah”, “uh”) and partial words are frequently used. This has an impact on the quality of the semantic graph, e.g., two totally semantically irrelevant sentences will still have a (small) nonzero similarity score if they both begin with the filler word “so”. Consequently, one vertex may be weakly connected to many other semantically unrelated vertices, and the f_{cut} function leads to this noise being accumulated to the point of becoming significant. This explains why in Table I we see f_{cut} (summation of summation) performing poor, $f_{facility}$ (summation of maximums) performing better, and f_{worst} (no summation) performing the best. On the other hand, with ASR outputs (Table II), filler and partial words can be “removed” or miss-recognized by the ASR engine’s imperfect output, which tends not to produce a consistent low-level similarity between many sentences. Therefore, Table II’s results are less affected by such noise.

With the ROUGE-1 graph construction method, words with low TF-IDF scores were removed prior to the actual computation of the similarity scores. The resulting graph is sparse and less noisy. As we can see in Table III and Table IV, all graph-based methods including PageRank perform better on both human transcripts and ASR outputs. Nevertheless, given the same semantic graph, submodular selection outperforms the recursive graph-based ranking algorithm, demonstrating the power of submodularity.

VI. CONCLUSION AND FUTURE WORK

Graph-based submodular selection is a simple and effective approach for extractive summarization. Experiments on ICSI

TABLE II
ROUGE-1 F-MEASURE RESULTS (%) FOR DIFFERENT WORD
COMPRESSION RATIOS FOR ASR OUTPUTS (ASR) ON BOTH DEV. SET AND
TEST SET WITH GRAPHS BASED ON COSINE SIMILARITY.

ASR.G-cosine.DEV.	ROUGE-1 F-Measure (%)				
	13%	14%	15%	16%	17%
Word comp. ratio					
MMR	62.59	63.60	64.32	64.80	65.03
ILP	62.59	63.99	65.04	65.45	65.44
PageRank-U	54.56	54.60	54.50	54.41	54.41
PageRank-F	62.21	62.21	62.22	61.97	61.57
PageRank-B	63.83	64.19	64.23	63.94	63.45
Submodular- $f_{facility}$	65.54	65.82	66.15	66.21	65.72
Submodular- f_{cut}	63.33	64.00	64.15	64.29	63.89
Submodular- f_{worst}	65.78	65.91	66.10	65.99	65.40
Submodular- $f_{penalty}$	66.71*	66.81*	66.71*	66.60*	65.90*
ASR.G-cosine.TEST	ROUGE-1 F-Measure (%)				
Word comp. ratio	13%	14%	15%	16%	17%
MMR	61.29	62.35	63.36	63.91	64.22
ILP	62.18	63.30	64.51	65.31	65.27
PageRank-U	56.01	56.17	56.35	56.38	56.33
PageRank-F	61.23	61.78	62.03	62.01	61.70
PageRank-B	61.96	62.50	62.94	62.93	62.88
Submodular- $f_{facility}$	64.74	65.35	65.65	65.86	65.43
Submodular- f_{cut}	63.04	63.72	64.25	64.29	64.05
Submodular- f_{worst}	64.25	64.93	65.06	64.81	64.48
Submodular- $f_{penalty}$	66.17*	66.60*	66.76*	66.61*	66.08*

TABLE III
ROUGE-1 F-MEASURE RESULTS (%) FOR DIFFERENT WORD
COMPRESSION RATIO FOR HUMAN TRANSCRIPTS (REF) ON BOTH DEV.
SET AND TEST SET WITH ROUGE SCORE BASED GRAPHS.

REF.G-ROUGE.DEV.	ROUGE-1 F-Measure (%)				
	13%	14%	15%	16%	17%
Word comp. ratio					
MMR	66.28	66.81	67.06	66.90	66.64
ILP	66.46	67.20	67.98	68.30	67.82
PageRank-U	68.69	69.24	69.24	69.12	68.92
PageRank-F	65.43	66.37	66.69	66.67	66.58
PageRank-B	69.00	69.27	69.63	69.30	68.85
Submodular- $f_{facility}$	68.53	69.16	69.29	69.32	68.83
Submodular- f_{cut}	69.21*	69.82*	69.82*	70.16*	69.89*
Submodular- f_{worst}	68.62	69.07	69.32	69.58	68.97
Submodular- $f_{penalty}$	68.75	69.17	69.01	69.02	68.97
REF.G-ROUGE.TEST	ROUGE-1 F-Measure (%)				
Word comp. ratio	13%	14%	15%	16%	17%
MMR	64.67	65.69	66.23	66.69	66.70
ILP	66.11	67.08	67.84	68.35	68.82
PageRank-U	67.98	69.15	69.69	69.73	69.59
PageRank-F	66.37	67.29	67.81	67.99	68.21
PageRank-B	67.30	68.02	68.40	68.73	68.50
Submodular- $f_{facility}$	67.53	68.65	69.03	69.31	68.87
Submodular- f_{cut}	68.00	69.04	69.94	70.27	70.16
Submodular- f_{worst}	67.75	68.82	69.20	69.60	69.61
Submodular- $f_{penalty}$	69.08*	69.65*	70.07*	70.50*	70.48*

meeting summarization tasks show that our approach is quite promising. Note that it is possible to further boost our performance by building a better graph using richer features (e.g. lexical, structural and discourse features as used in [30][31]). Also, although proposed for meeting summarization, our approach is applicable to general extractive text summarization tasks. We plan to explore such possibilities in future work.

Acknowledgments: We thank Yang Liu for a helpful discussion. This work is supported by an ONR MURI grant (No. N000140510388), the Companions project (IST programme under EC grant IST-FP6-034434), and the EU 6th FWP IST Integrated Project AMI (FP6-506811).

TABLE IV

ROUGE-1 F-MEASURE RESULTS (%) FOR DIFFERENT WORD COMPRESSION RATIO FOR ASR OUTPUTS (ASR) ON BOTH DEV. SET AND TEST SET WITH ROUGE SCORE BASED GRAPHS.

ASR.G-ROUGE.DEV.	ROUGE-1 F-Measure (%)					
	Word comp. ratio	13%	14%	15%	16%	17%
MMR		62.59	63.60	64.32	64.80	65.03
ILP		62.59	63.99	65.04	65.45	65.44
PageRank-U		64.51	65.16	65.20	65.36	64.98
PageRank-F		63.36	64.13	64.42	64.33	64.15
PageRank-B		65.03	65.43	65.76	65.78	65.43
Submodular- f_{facility}		64.84	65.51	65.72	65.52	65.07
Submodular- f_{cut}		65.85	66.13*	66.04*	66.02*	65.64*
Submodular- f_{worst}		64.42	65.08	65.47	65.39	65.07
Submodular- f_{penalty}		65.94*	65.96	65.94	65.82	65.48
ASR.G-ROUGE.TEST	ROUGE-1 F-Measure (%)					
Word comp. ratio	13%	14%	15%	16%	17%	
MMR		61.29	62.35	63.36	63.91	64.22
ILP		62.18	63.30	64.51	65.31	65.27
PageRank-U		64.11	64.95	65.49	65.55	65.45
PageRank-F		63.08	63.82	64.54	64.68	64.61
PageRank-B		64.77	65.49	65.62	65.96	65.56
Submodular- f_{facility}		64.35	65.46	65.98	65.90	65.73
Submodular- f_{cut}		64.97	65.69	66.38	66.59	66.52
Submodular- f_{worst}		64.15	65.23	65.88	66.02	65.80
Submodular- f_{penalty}		65.53*	66.51*	66.96*	67.05*	67.19*

REFERENCES

- [1] R. McDonald, "A study of global inference algorithms in multi-document summarization," *Lecture Notes in Computer Science*, vol. 4425, p. 557, 2007.
- [2] H. Dang, "Overview of DUC 2005," in *Proceedings of the Document Understanding Conference*, 2005.
- [3] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proc. of SIGIR*, 1998.
- [4] A. Krause, H. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, pp. 2761–2801, 2008.
- [5] D. Gillick, K. Riedhammer, B. Favre, and D. Hakkani-Tür, "A global optimization framework for meeting summarization," in *Proc. of ICASSP*, 2009.
- [6] R. Mihalcea, P. Tarau, and E. Figa, "PageRank on semantic networks, with application to word sense disambiguation," in *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, 2004.
- [7] L. Lovasz, "Submodular functions and convexity," *Mathematical programming-The state of the art*, (eds. A. Bachem, M. Grottschel and B. Korte) Springer, pp. 235–257, 1983.
- [8] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [9] U. Feige, "A threshold of $\ln n$ for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.
- [10] G. Cornuejols, M. Fisher, and G. Nemhauser, "On the uncapacitated location problem," in *Studies in Integer Programming: Proceedings of the Institute of Operations Research Workshop*, vol. 1. North Holland, 1977, pp. 163–177.
- [11] L. Shapley, "Cores of convex games," *International Journal of Game Theory*, vol. 1, no. 1, pp. 11–26, 1971.
- [12] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time," *Journal of Combinatorial Theory, Series B*, vol. 80, no. 2, pp. 346–355, 2000.
- [13] S. Iwata, L. Fleischer, and S. Fujishige, "A combinatorial strongly polynomial algorithm for minimizing submodular functions," *Journal of the ACM*, vol. 48, no. 4, pp. 761–777, 2001.
- [14] M. Narasimhan and J. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," in *Proc. Conf. Uncertainty in Artificial Intelligence*. Edinburgh, Scotland: Morgan Kaufmann Publishers, July 2005.

- [15] M. Narasimhan and J. Bilmes, "Local search for balanced submodular clusterings," in *Twentieth International Joint Conference on Artificial Intelligence (IJCAI07)*, Hyderabad, India, January 2007.
- [16] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [17] A. R. Krause, "Optimizing sensing: Theory and applications," Ph.D. dissertation, Carnegie Mellon University, 2008.
- [18] G. Erkan and D. Radev, "LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.
- [19] R. Mihalcea and P. Tarau, "TextRank: bringing order into texts," in *Proceedings of EMNLP*, Barcelona, Spain, 2004.
- [20] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.
- [21] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of the ACL 2004 (companion volume)*, 2004.
- [22] H. Lin and J. A. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Proc. of Interspeech*, Brighton, UK, September 2009.
- [23] A. Janin, D. Baron, and J. E. et al., "The ICSI meeting corpus," in *Proc. of ICASSP*, 2003.
- [24] G. Murray, S. Renals, and J. Carletta, "Extractive summarization of meeting recordings," in *Proc. of Interspeech*, 2005.
- [25] S. Xie, B. Favre, D. Hakkani-Tür, and Y. Liu, "Leveraging sentence weights in a concept-based optimization framework for extractive meeting summarization," in *Proc. of Interspeech*, 2009.
- [26] Q. Zhu, A. Stolcke, B. Chen, and N. Morgan, "Using MLP features in SRI's conversational speech recognition system," in *Proc. of Interspeech*, 2005.
- [27] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004.
- [28] J. J. Zhang, H. Y. Chan, and P. Fung, "Improving lecture speech summarization using rhetorical information," in *Proc. of ASRU*, 2007.
- [29] X. Zhu and G. Penn, "Summarization of spontaneous conversations," in *Proc. of Interspeech*, 2006.
- [30] M. Galley, "A skip-chain conditional random field for ranking meeting utterances by importance," in *Proc. of EMNLP*, 2006.
- [31] S. Xie, Y. Liu, and H. Lin, "Evaluating the effectiveness of features and sampling in extractive meeting summarization," in *Proc. of IEEE Spoken Language Technology (SLT)*, 2008.

APPENDIX

Theorem 3. $\forall i, g_i(S) = \max_{j \in S} w_{i,j}$ is submodular and non-decreasing.

Proof: For $k \in V \setminus S$,

$$g_i(S \cup k) - g_i(S) = \max \left\{ 0, w_{i,k} - \max_{j \in S} w_{i,j} \right\} \geq 0 \quad (14)$$

so $g_i(S)$ is non-decreasing. Also, for $R \subseteq S$, we have

$$g_i(S \cup k) - g_i(S) \leq g_i(R \cup k) - g_i(R) \quad (15)$$

using the above, indicating that $g_i(S)$ is submodular. ■

Theorem 4. f_{penalty} is submodular.

Proof: Since $\sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j}$ is submodular and λ is non-negative, we only need to show the penalty term $g(S) = -\sum_{i,j \in S: i \neq j} w_{i,j}$ is submodular. For $k \in V \setminus S$, we have

$$\rho_k(S) = g(S \cup \{k\}) - g(S) = - \left(\sum_{i \in S \setminus \{k\}} w_{i,k} + \sum_{j \in S \setminus \{k\}} w_{k,j} \right)$$

Since weights $w_{i,j}$ are non-negative, $\rho_k(S) \leq \rho_k(R)$ for $R \subseteq S$. ■