5 Appendix

5.1 Proof of Lemma 3

Proof. Firstly, we have the following inequality.

$$\begin{aligned} f(x|v) &= f(x+u|v) - f(u|v+x) \\ &= f(x|u+v) + f(u|v) - f(u|v+x) \\ &\leq f(x|u) + f(u|v) - f(u|v+x). \end{aligned} \tag{15}$$

The first two equalities follow from the definition of marginal gain, while the inequality is due to submodularity. Following the definition of w_{uv} in Eq. (3), we have

$$\begin{aligned} w_{vx} &= f(x|v) - f(v|V-v) \\ &\leq f(x|u) + f(u|v) - f(u|v+x) - f(v|V-v) \\ &\leq [f(x|u) - f(u|V-u)] + [f(u|v) - f(v|V-v)] \\ &= w_{ux} + w_{vu}. \end{aligned} \tag{16}$$

The first inequality is due to Eq. (15), and the second inequality is via submodularity. □

5.2 Proof of Proposition 1

Proof. Define a set A_u for each $u \in V'$ such that $A_u = \{v \in V : w_{uv} \leq \epsilon\}$. Note $u \in A_u$ because $w_{uu} = -f(u|V\backslash u) \leq 0 \leq \epsilon$ and hence $V' \subseteq \cup_{u \in V'} A_u$. The objective function h in Eq. (8) can be written as

$$\begin{aligned} h(V') &= |\{v \in V\backslash V' : w_{V'v} \leq \epsilon\}| \\ &= |\{v \in V\backslash V' : \exists x \in V' : w_{xv} \leq \epsilon\}| \\ &= \left| \left(\bigcup_{u \in V'} A_u \right) \setminus V' \right| = \left| \bigcup_{u \in V'} A_u \right| - |V'|, \end{aligned} \tag{17}$$

where $f_{SC}(V') = \left|\bigcup_{u \in V'} A_u\right|$ is the simple set cover function [11], which is monotone non-decreasing submodular, and $-|V'|$ is a monotone decreasing modular (negative cardinality) function. Because the sum of a submodular function and a modular function is still submodular, the objective in Eq. (8) is non-monotone submodular. □

5.3 Proof of Theorem 1

Proof. Recall that $u_v^* \in \text{argmin}_{u \in V^*} w_{uv}$ is the tail node of an edge with the minimal weight over all edges from elements in V^* to head v. Since $|V^*| \geq k$, the greedy algorithm on V^* will run for k steps and select k elements. We use S_i to denote the solution set at the beginning of the i^{th} step, let $u_i \in \text{argmax}_{x \in V^* \backslash S_i} f(x|S_i)$ be the selected element in this step. In addition, let $v_i = \text{argmax}_{x \in V \backslash S_i} f(x|S_i)$ be the unfettered greedy choice at step i. Then we have the following:

$$\begin{aligned} f(v_i|S_i) \quad &\leq f(u_i|S_i) + \min_{x \in V^*} w_{xv_i|S} \\ &\leq f(u_i|S_i) + \min_{x \in V^*} w_{xv_i} \\ &= f(u_i|S_i) + w_{u^*_{v_i} v_i} \\ &\leq f(u_i|S_i) + \epsilon. \end{aligned} \tag{18}$$

The first inequality is by Eq. (7), the second inequality is due to Lemma 1, while the last inequality comes from the definition of problem Eq. (8). Hence, for arbitrary i, we have

$$\begin{aligned} f(S^*) \quad &\leq f(S_i \cup S^*) \\ &\leq f(S_i) + \sum_{x \in S^* \backslash S_i} f(x|S_i) \\ &\leq f(S_i) + \sum_{x \in S^*} f(x|S_i) \\ &\leq f(S_i) + k \max_{x \in V} f(x|S_i) \\ &= f(S_i) + k f(v_i|S_i) \\ &\leq f(S_i) + k\left[f(u_i|S_i) + \epsilon\right] \\ &= f(S_i) + k\left[f(S_{i+1}) - f(S_i) + \epsilon\right]. \end{aligned} \tag{19}$$

The first inequality uses monotonicity of $f(\cdot)$, while the second one is due to submodularity. The third inequality is due to the non-negativity of $f(\cdot)$. The fourth inequality is due to the maximal greedy selection rule for the greedy algorithm on the original ground set V. The fifth inequality is the result of applying Eq. (18). The last equality is due to the greedy selection rule $S_{i+1} = u_i \cup S_i$ for the greedy algorithm on the reduced ground set V^*. Rearranging Eq. (19) yields

$$[f(S^*) - k\epsilon] - f(S_i) \leq k[f(S_{i+1}) - f(S_i)] \tag{20}$$

Let

$$\delta_i = [f(S^*) - k\epsilon] - f(S_i), \tag{21}$$

then the rearranged inequality equals to

$$\delta_i \leq k[\delta_i - \delta_{i+1}], \tag{22}$$

Since $\delta_i - \delta_{i+1} \geq 0$, this equals to

$$\delta_{i+1} \leq \left(1 - \frac{1}{k}\right) \delta_i. \tag{23}$$

Since in total k elements are selected by the greedy algorithm, applying Eq. (23) from $i = 0$ to $i = k$ yields

$$\delta_k \leq \left(1 - \frac{1}{k}\right)^k \delta_0 \leq e^{-1} \delta_0. \tag{24}$$

By using the definition of δ_i in Eq. (21), the above inequality leads to

$$f(S') = f(S_k) \geq \left(1 - e^{-1}\right) \left(f(S^*) - k\epsilon\right). \tag{25}$$

This completes the proof. □

5.4 Proof of Lemma 4

Proof. The proof follows from Lemma 3 and our assumption to u.

$$\begin{aligned} w_{uv} \leq & w_{uu^*_v} + w_{u^*_v v} \\ =& f(v|u^*_v) + f(u^*_v|u) - f(u^*_v|V\backslash u^*_v) - f(u|V\backslash u) \\ =& f(v + u^*_v) + f(u + u^*_v) - f(u^*_v) - f(u) \\ & - f(u^*_v|V\backslash u^*_v) - f(u|V\backslash u) \\ \leq& 2f(v + u^*_v) - f(u^*_v) - f(u) \\ & - f(u^*_v|V\backslash u^*_v) - f(u|V\backslash u) \\ =& 2\left[f(v|u^*_v) - f(u^*_v|V\backslash u^*_v)\right] \\ & + \left[f(u^*_v) + f(u^*_v|V\backslash u^*_v) - f(u) - f(u|V\backslash u)\right] \\ \leq& 2w_{u^*_v v}. \end{aligned}$$

The first inequality is due to Lemma 3. The second inequality is because $f(u + u^*_v) \leq f(v + u^*_v)$ which follows from

$u \in P(u_v^*)$. The third inequality is due to $u \in Q(u_v^*)$. □

5.5 Proof of Proposition 2

Proof. Recall V^* is the optimal solution of problem in Eq. (8). Due to the definition of $|V|/(cK)$-NN ball, we have

$$\forall v \in V_{u^*} \backslash B\left(u^*, |V|/(cK)\right), f(u+u^*) \leq f(v+u^*). \tag{26}$$

Hence, $u \in P(u_v^*) \cap Q(u_v^*)$. By using Lemma 4, we have

$$w_{uv} \leq 2w_{u^*v}. \tag{27}$$

This completes the proof. □

5.6 Proof of Proposition 3

Proof. According to Proposition 2, for each $u^* \in V^*$, if one $u \in B\left(u^*, |V|/(cK)\right) \cap Q(u^*)$ is sampled into U in some iteration of Algorithm 1, then any item v outside the ball satisfies

$$\begin{aligned} w_{Uv} &= \min_{x \in U} w_{xv} \leq w_{uv} \\ &\leq 2w_{u_v^*v} = 2w_{V^*v}. \end{aligned}$$

Hence, one element u fulfilling $w_{Uu} \geq 2w_{V^*u}$ in the complement set must be contained in least one of the K $|V|/(cK)$-NN balls whose centers are the K elements in V^*. Therefore, the total number of such u is at most $|V|/c = K \times |V|/(cK)$, the maximal number of elements in all the K $|V|/(cK)$-NN balls. □

5.7 Proof of Proposition 4

Proof. We consider V_i, set V at the beginning of the i^{th} iteration, and V_{i-1}, set V right before the removal step of the previous iteration. According to the pruning amount $1 - 1/\sqrt{c}$:

$$|V_i| = 1/\sqrt{c}|V_{i-1}|. \tag{28}$$

Since Proposition 3 indicates

$$|\{u \in V_i : w_{Uu} \geq 2w_{V^*u}\}| \leq \frac{|V_{i-1}|}{c}, \tag{29}$$

we have

$$\begin{aligned} &|\{v \in V_i : w_{Uv} \leq 2w_{V^*v}\}| \\ =&|V_i| - |\{u \in V_i : w_{Uu} \geq 2w_{V^*u}\}| \\ \geq&\frac{1}{\sqrt{c}}|V_{i-1}| - \frac{1}{c}|V_{i-1}| \\ =&\left(1 - \frac{1}{\sqrt{c}}\right) \times (\frac{1}{\sqrt{c}})|V_{i-1}| \\ =&\left(1 - \frac{1}{\sqrt{c}}\right)|V_i|. \end{aligned}$$

Because the above result is correct for arbitrary i, it completes the proof. □

5.8 Proof of Lemma 5

Proof. According to Proposition 4, after removal, all the elements in $\{v \in V : w_{Uv} > 2w_{V^*v}\}$ are retained in V'. So none of them is in $V \backslash V'$.

According to Proposition 2, if for each $u^* \in V^*$ at least one alternate $u \in B\left(u^*, |V|/(cK)\right) \cap Q(u^*)$ is sampled and added into U, $\forall v \in V$, we have $w_{V'v} \leq 2w_{V^*v}$. This completes the proof. □

5.9 Proof of Proposition 5

Proof. According to the assumption and definition of $Q(u^*)$ in Lemma 4, $\forall u \in U$,

$$\Pr\left[u \in Q(u^*) | u \in B\left(u^*, |V|/(cK)\right)\right] \geq q. \tag{30}$$

In addition, the probability for that an uniform sample u is inside the $|V|/(cK)$-NN ball $B\left(u^*, |V|/(cK)\right)$ of u^* is

$$\Pr\left(u \in B\left(u^*, |V|/(cK)\right)\right) = \frac{1}{cK}. \tag{31}$$

Combining the two probabilities, we have

$$\Pr\left(u \notin B\left(u^*, |V|/(cK)\right) \cap Q(u^*)\right) \leq 1 - \frac{q}{cK}. \tag{32}$$

Since $r = O(cK) = pcK$, among the $r \log n = pcK \log n$ samples of U in one iteration, for one specific u^*, the probability that no sample belongs to $B\left(u^*, |V|/(8K)\right) \cap Q(u^*)$ is

$$\begin{aligned} &\Pr\left(U \cap \left(B\left(u^*, |V|/(8K)\right) \cap Q(u^*)\right) = \emptyset\right) \\ &\leq \left(1 - \frac{q}{cK}\right)^{r \log n} = \left(1 - \frac{q}{cK}\right)^{pcK \log n} \leq n^{-qp}. \end{aligned}$$

Note there are K items in V^*, and there will be at most $\log_{\sqrt{c}} n$ iterations. By union bound, the failure probability that no $u \in B(u^*, |V|/(cK)) \cap Q(u^*)$ is sampled and added into U for at least one $u^* \in V^*$ in at least one iteration of Algorithm 1 is at most

$$K \times n^{-qp} \times \log_{\sqrt{c}} n \leq n^{1-qp} \log_{\sqrt{c}} n. \tag{33}$$

□

5.10 Discussion of q

In Proposition 5 and Theorem 2, the failure probability is small when q is large. Here q is a lower bound of conditional probability

$$\Pr\left[u \in Q(u^*) | u \in B\left(u^*, |V|/(cK)\right)\right] \geq q. \tag{34}$$

The conditional probability describes the probability of

$$f(u) + f(u|V\backslash u) \geq f(u^*) + f(u^*|V\backslash u^*) \tag{35}$$

for item u inside the ball $B\left(u^*, |V|/(cK)\right)$.

In §3.4, we mentioned a weighted resampling method that samples probe items from uniformly sampled items (with weight of item u proportional to $f(u) + f(u|V\backslash u)$) to achieve a sufficiently large q. The effectiveness of this method has been broadly demonstrated in our experiments.

However, in theory it is hard to directly quantify how large the conditional probability is because it is data dependent. Nevertheless, we can discuss the possible values of its lower bound q in two special cases and the general case. The discussion shows that by increasing the size of ball $B\left(u^*, |V|/(cK)\right)$ (via reducing c) we can achieve large and more interpretable q.

In the following, we will use $\mathbb{B}$ to denote $B\left(u^*, |V|/(cK)\right)$ analysis for simplicity. Note each $u^* \in V^*$ is (unknown) constant, and all quantities related to u^* need to be treated as constant as well.

Firstly, we will discuss two special cases in which q can be chosen as $q = \Pr\left[u \in Q(u^*)\right]$ if ϵ is sufficiently small and $\mathbb{B}$ is sufficiently large. Let

$$R \triangleq \max_{v \in \mathbb{B}} f(v + u^*) \tag{36}$$

to be $f(v + u^*)$ for any item v on the boarder of the $|V|/(cK)$-NN ball $\mathbb{B}$, then $u \in \mathbb{B}$ is equivalent to

$$f(u|u^*) \le R - f(u^*). \tag{37}$$

When 1) $\epsilon \le R - f(u^*)$; or 2) $\forall u \in V_{u^*}$[5], $f(u) \le R - f(u^*)$, $f(u|u^*) \le R - f(u^*)$ is always true, because 1) in the first case, $f(u|u^*) \le \epsilon$ holds true due to $u^* \in V^*$, the optimal solution to Eq. (8); and 2) in the second case, $f(u) \le f(u|u^*)$ holds true due to submodularity. In both cases, the condition $u \in \mathbb{B}$ is always true and can be removed. Therefore,

$$\Pr\left[u \in Q(u^*) | u \in \mathbb{B}\right] = \Pr\left[u \in Q(u^*)\right]. \tag{38}$$

Case 1) holds true if ϵ is sufficiently small and/or R is sufficiently large, while Case 2) holds true if R is sufficiently large. make them true, we can reduce ϵ when defining the problem in Eq. (8), and increase R by increasing the size of $\mathbb{B}$ (via reducing c).

Secondly, in the general case, the condition $f(u) \le R - f(u^*)$ in Case 2) holds true only for some $u \in V_{u^*}$. Let the set of these items to be $V^0_{u^*}$. For the rest items $u \in V_{u^*} \backslash V^0_{u^*}$, we have $f(u) > R - f(u^*)$, so

$$\begin{aligned}&f(u|V\backslash u) \ge 2f(u^*) + f(u^*|V\backslash u^*) - R \Rightarrow\\ &f(u) + f(u|V\backslash u) \ge f(u^*) + f(u^*|V\backslash u^*) \Leftrightarrow\\ &u \in Q(u^*).\end{aligned}$$

Hence, let constant $a \triangleq 2f(u^*) + f(u^*|V\backslash u^*)$, for $u \in V_{u^*} \backslash V^0_{u^*}$,

$$\Pr\left[u \in Q(u^*) | u \in \mathbb{B}\right] \ge \Pr\left[f(u|V\backslash u) \ge a - R | u \in \mathbb{B}\right]. \tag{39}$$

Combining analysis for u in the two sets, for $u \in V_{u^*}$,

$$\begin{aligned}&\Pr\left[u \in Q(u^*) | u \in \mathbb{B}\right] \ge \Pr\left[u \in Q(u^*)\right] \times \frac{|V^0_{u^*}|}{|V_{u^*}|} +\\ &\Pr\left[f(u|V\backslash u) \ge a - R | u \in \mathbb{B}\right] \times \left(1 - \frac{|V^0_{u^*}|}{|V_{u^*}|}\right).\end{aligned}$$

Therefore, in the general case, q can be chosen as the right hand side of the above inequality. The two terms in the right hand side increases when increasing R. So a sufficiently large q can be achieved by increasing the size of $\mathbb{B}$ (via reducing c again).

When treating q as an unknown constant, the success probability increases when increasing c, as discussed after Theorem 2. However, when the value of q is chosen as suggested above, the behavior of the failure probability $n^{1-qp} \times \log_{\sqrt{c}} n$ when changing c becomes more complex. According to the analysis above, reducing c increases q so decreases the first factor n^{1-qp}, but it increase the second factor $\log_{\sqrt{c}} n$. This makes the failure probability increases in some intervals of c but decreases in others, and finding a good c is critical and worthy studying in future works.

5.11 Proof of Theorem 2

Proof. Firstly, since $r \log n = pcK \log n$ elements are selected into V' per iteration, and the number of iterations is $\log_{\sqrt{c}} n$, so the size of V' is

$$|V'| = pcK \log n \times \log_{\sqrt{c}} n = (pc/\log_{\sqrt{c}})K \log^2 n. \tag{40}$$

Secondly, combing the results of Lemma 5 and failure probability $n^{1-qp} \log_{\sqrt{c}} n$ in Proposition 5, we have: with success probability $1 - n^{1-qp} \log_{\sqrt{c}} n$, $\forall v \in V \backslash V'$, $w_{V'v} \le 2w_{V^*v}$.

Thirdly, since $w_{V'v} \le 2w_{V^*v}$, we replace $w_{u^*_{v_i} v_i}$ with $2w_{u^*_{v_i} v_i}$ in Eq. (18), the rest proof of Theorem 1 leads to

$$f(S') \ge \left(1 - e^{-1}\right)\left(f(S^*) - 2k\epsilon\right). \tag{41}$$

This completes the proof. □

5.12 Comparison to Pruning method in [28]

We provide a theoretical comparison to the pruning method in [28], which removes all elements whose singular gain $f(v) \le f(u_k|V|u_k)$, where u_k is the elements with the k^{th} largest value of $f(u|V\backslash u)$ in V. Comparing to this method, SS removes elements by thresholding of edge weights with form $w_{uv} = f(v|u) - f(u|V\backslash u)$. It can be seem that w_{uv} involves both the global information $f(u|V\backslash u)$ used in [28] and local (mutual) relationship $f(v|u)$. We show below that this helps to remove more elements.

We compare [28] with a simpler thresholding strategy $w_{uv} \le 0$ which removes less elements than Algorithm 1, because each iteration of SS equals using a thresholding strategy $w_{uv} \le \tau$ with $\tau > 0$ (it is easy to guarantee $\tau > 0$ by tuning c in Algorithm 1). The simpler strategy keeps u and removes v if $w_{uv} \le 0$, which equals to $f(v|u) \le f(u|V\backslash u)$. If the reduced ground set is $V^{'}$, the removed set of elements is

$$\begin{aligned}V\backslash V^{'} = \{&v \in V | f(v|u_v) - f(u_v|V\backslash u_v) \le 0,\\ &u_v \in \text{argmin}_{u \in V'}\left[f(v|u) - f(u|V\backslash u)\right]\}.\end{aligned} \tag{42}$$

The simpler strategy is lossless on achieved objective value, because w_{uv} is the loss caused by this pruning action, and a non-positive w_{uv} will keep the achieved objective value the same. Hence, the approximation bound of greedy or lazy greedy on the reduced ground set is the same as the one achieved on the original ground set.

[5] Definition of V_{u^*} was given in Proposition 2

Pruning method in [28] has the same guarantee (according to Lemma 1 in [28]). Its reduced ground set can be represented as

$$V\backslash V^{'} = \{v \in V | f(v) - f(u_k|V\backslash u_k) \leq 0\}. \tag{43}$$

The thresholding rules (the inequality) in (42) and (43) have similar forms now. To discuss their relationship, we consider two cases in the following.

When $u_k \in V^{'}$, we have

$$\begin{aligned} & f(v|u_v) - f(u_v|V\backslash u_v) \\ = \; & \min_{u\in V^{'}} [f(v|u) - f(u|V\backslash u)] \\ \leq \; & f(v|u_k) - f(u_k|V\backslash u_k) \\ \leq \; & f(v) - f(u_k|V\backslash u_k). \end{aligned} \tag{44}$$

So the removed set in (43) is a subset of the removed set in (42). This indicates that the simpler strategy based on w_{uv} can remove more elements.

When $u_k \notin V^{'}$, we can always find a $u_k^{'} \in V^{'}$ as an alternative of u_k such that

$$f(u_k|u_k^{'}) - f(u_k^{'}|V\backslash u_k^{'}) \leq 0, \tag{45}$$

so we have

$$f(u_k|V\backslash u_k) \leq f(u_k|u_k^{'}) \leq f(u_k^{'}|V\backslash u_k^{'}). \tag{46}$$

Therefore,

$$\begin{aligned} & f(v|u_v) - f(u_v|V\backslash u_v) \\ = \; & \min_{u\in V^{'}} [f(v|u) - f(u|V\backslash u)] \\ \leq \; & f(v|u_k^{'}) - f(u_k^{'}|V\backslash u_k^{'}) \\ \leq \; & f(v) - f(u_k|V\backslash u_k). \end{aligned} \tag{47}$$

So the removed set in (43) is a subset of the removed set in (42). This, again, indicates that the simpler strategy based on w_{uv} can remove more elements.

Combining the two cases, we can conclude that SS can remove more elements than pruning method in [28].

5.13 Comparison to Parallel Methods

Parallelization can be used to accelerate the computations within each iteration of SS. Because it only computes some pairwise edge weights. So parallelization tricks to compute pairwise measures can be used in our case as well (and is a widely used method). But in our experiments we sequentially compute the edge weights to fairly compare with the other methods. Moreover, our main contribution is not a parallel method. Our experiment do not use or require parallelization. In fact, SS in Algorithm 1 prunes elements sequentially subset by subset: which subset of elements to remove relies on which elements have been removed before.

Our major contribution is the graph defined by a new form of edge weight, and the graph based sequentially pruning method: it does not rely on any parallelization. The idea is very different from previous parallel submodular maximization methods. Our methods are entirely complementary with those methods, so we do not believe it is necessary to give a detailed empirical comparison to parallel submodular maximization.

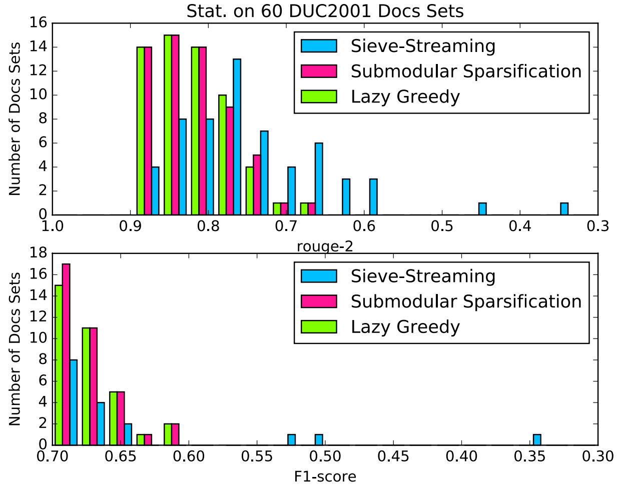

Figure 7: Statistics of relative utility $f(S)/f(S_{greedy})$, ROUGE-2 score and F1-score on topic based news summarization results of 60 document sets from DUC2001 training and test set, comparing to 400-word human generated summary.

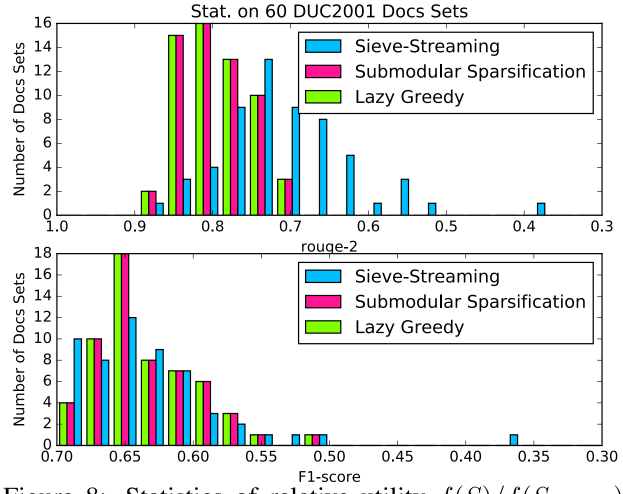

Figure 8: Statistics of relative utility $f(S)/f(S_{greedy})$, ROUGE-2 score and F1-score on topic based news summarization results of 60 document sets from DUC2001 training and test set, comparing to 200-word human generated summary.

5.14 Experiments on DUC2001 News Summarization

We also observe similar result on DUC 2001 corpus, which are composed of two datasets. The first one includes 60 sets of documents, each is selected by a NIST assessor because the documents in a set are related to a same topic. The assessor also provides four human generated summary of word count $400, 200, 100, 50$ for each set. In Figure 7 and Figure 8, we report the statistics to ROUGE-2 and F1-score of summaries of the same size generated by different algorithms. The second dataset is composed of four document sets associated with four topics. We report the detailed results in Table 1. Both of them show submodular sparsification can achieve similar performance as greedy algorithm,

Table 1: Performance of Lazy greed, sieve-streaming, and submodular sparsification on four topic summarization datasets from DUC 2001. For each topic, the machine generated summary is compared to four human generated ones of word count from 50 to 400.

Algorithm	words	Daycare		Healthcare		Pres92		Robert Gates	
		ROUGE2	F1	ROUGE2	F1	ROUGE2	F1	ROUGE2	F1
Lazy Greedy	400	0.836	0.674	0.845	0.686	0.885	0.686	0.849	0.734
	200	0.813	0.615	0.811	0.632	0.842	0.623	0.788	0.682
	100	0.766	0.542	0.753	0.605	0.618	0.420	0.715	0.621
	50	0.674	0.484	0.765	0.539	0.602	0.341	0.631	0.514
Sieve-Streaming	400	0.825	0.687	0.814	0.711	0.827	0.710	0.798	0.745
	200	0.789	0.627	0.782	0.675	0.670	0.659	0.691	0.688
	100	0.747	0.542	0.658	0.597	0.414	0.443	0.632	0.620
	50	0.607	0.475	0.681	0.551	0.413	0.345	0.553	0.477
SS	400	0.837	0.674	0.845	0.686	0.883	0.685	0.849	0.734
	200	0.813	0.615	0.811	0.632	0.842	0.623	0.788	0.682
	100	0.766	0.542	0.753	0.605	0.617	0.420	0.715	0.621
	50	0.674	0.484	0.765	0.539	0.602	0.341	0.631	0.514

whereas outperforms sieve-streaming.

5.15 Experiments on Video Summarization

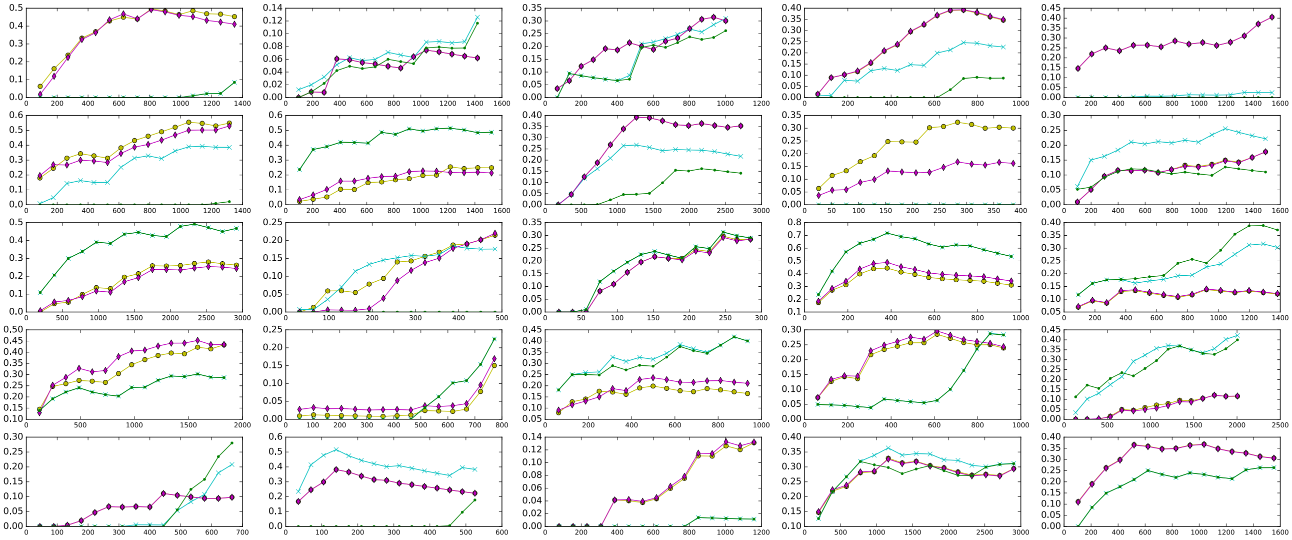

Figure 9: F1-score of the summaries generated by lazy greedy ("●"), sieve-streaming ("×"), submodular sparsification ("♦") and the first 15% frames ("·") comparing to reference summaries of different sizes between $[0.02|V|, 0.32|V|]$ based on ground truth score (voting from 15 users) on 25 videos from SumMe. Each plot associates with a video.

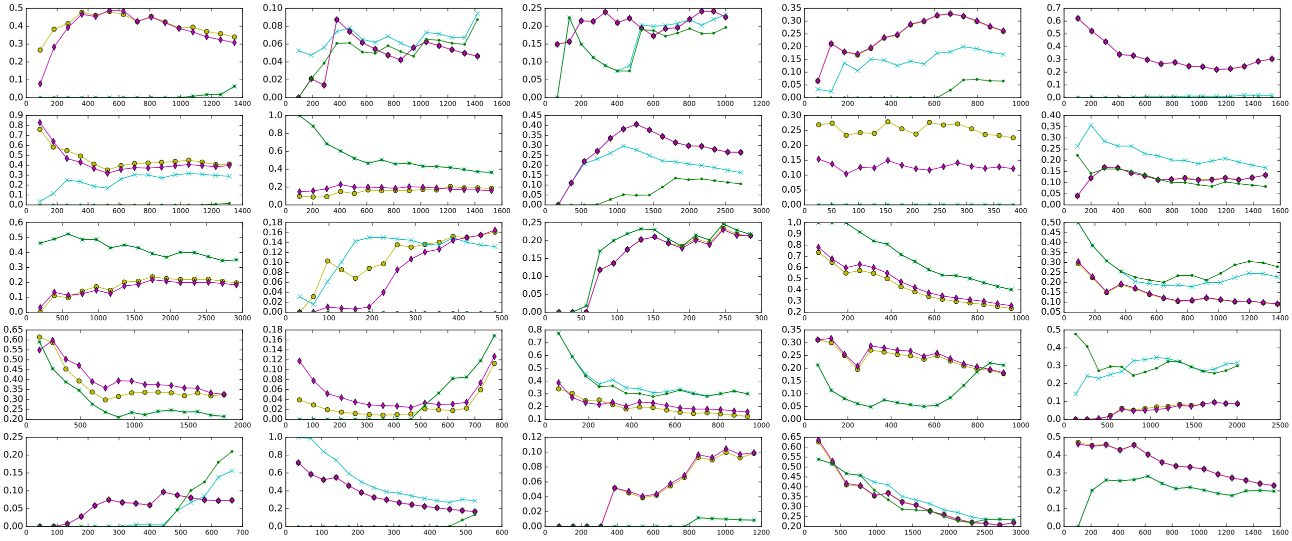

Figure 10: Recall of the summaries generated by lazy greedy ("●"), sieve-streaming ("×"), submodular sparsification ("♦") and the first 15% frames ("·") comparing to reference summaries of different sizes between $[0.02|V|, 0.32|V|]$ based on ground truth score (voting from 15 users) on 25 videos from SumMe. Each plot associates with a video.

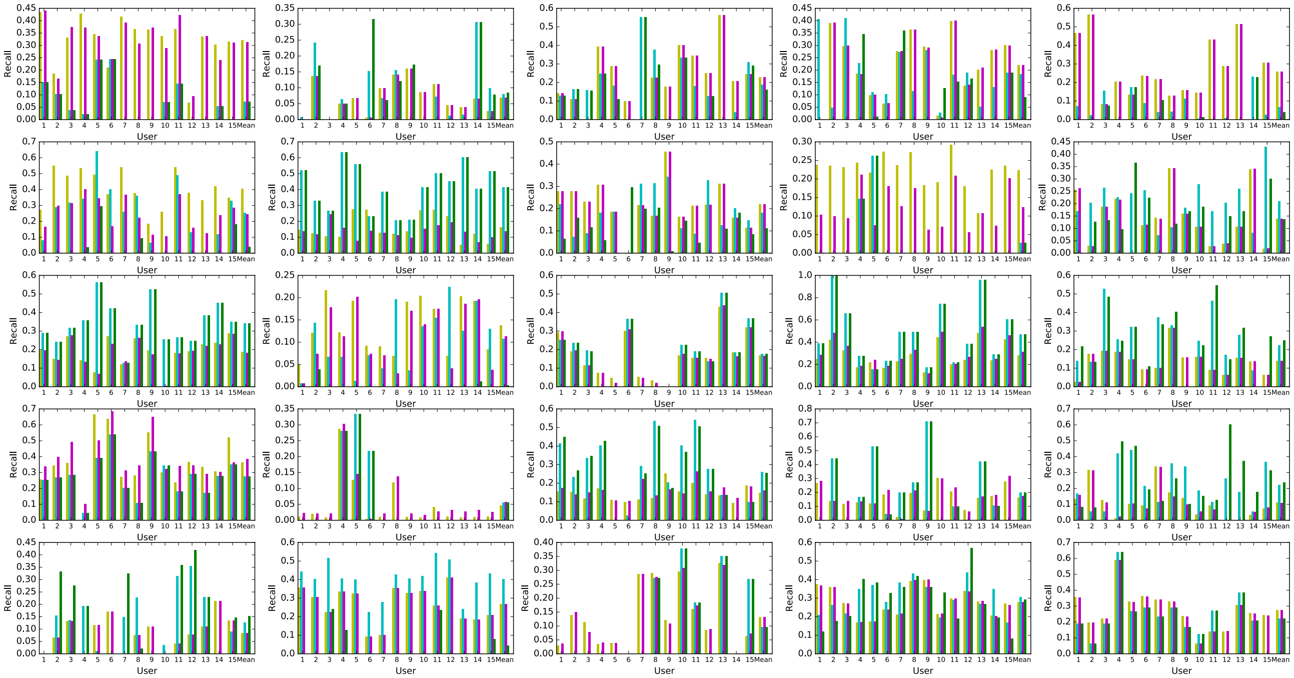

Figure 11: Recall of the summaries generated by greedy (yellow bar), sieve-streaming (cyan bar), SS (magenta bar) and the first 15% frames (green bar) comparing to reference summaries from 15 users on 25 videos from SumMe dataset. Each plot associates with a video.

Table 2: Information of SumMe dataset and time cost (CPU seconds) of different algorithms.

Video	#frames	$\|V'\|$	Lazy Greedy	Sieve-streaming	SS
Air Force One	4494	1031	907.3712	3.9182	71.4521
Base jumping	4729	1074	164.1434	5.5865	84.6877
Bearpark climbing	3341	1038	177.8583	3.7311	48.0415
Bike polo	3064	866	96.5305	3.9578	36.4832
Bus in rock tunnel	5131	1387	505.7766	6.0088	125.8121
Car over camera	4382	1396	146.9416	5.3323	69.6157
Car railcrossing	5075	1210	852.1686	5.2265	96.2396
Cockpit landing	9046	2292	669.8063	12.3186	212.7866
Cooking	1286	200	30.0717	1.2868	5.7096
Eiffel tower	4971	1647	304.2690	5.4755	86.5552
Excavators river crossing	9721	1971	1507.3028	13.8139	284.5136
Fire Domino	1612	464	34.2871	1.8814	9.9833
Jumps	950	308	15.0508	0.9055	4.8719
Kids playing in leaves	3187	986	221.4644	3.4660	41.1956
Notre Dame	4608	1136	169.1235	5.1406	72.9076
Paintball	6096	1664	763.3255	6.7853	128.1723
Paluma jump	2574	727	210.8670	2.5342	26.7430
Playing ball	3120	697	132.7437	3.2250	32.3198
Playing on water slide	3065	778	111.7358	3.4088	30.4131
Saving dolphines	6683	1860	435.0732	7.3322	121.5891
Scuba	2221	775	45.6177	2.5213	18.4227
St Maarten Landing	1751	628	19.0717	2.8701	12.4074
Statue of Liberty	3863	1223	160.7075	4.0164	55.7420
Uncut evening flight	9672	3324	718.7015	14.6717	208.8540
Valparaiso downhill	5178	1438	428.3941	6.0002	154.5902