Appendices

A Submodularity of f and c

Remember that f and c are defined on $\mathcal{P}$ as

$$f(X) \coloneqq g(V_X), \qquad c(X) \coloneqq \sum_{v \in V_X} \ell_v,$$

where $V_X \coloneqq \bigcup_{p \in X} V_p$; $V_p \subseteq V$ is a vertex subset that is included in path $p \in \mathcal{P}$.

We first see that f is a submodular function. Let $X \subseteq Y$ and $p \notin Y$, then f satisfies the diminishing return property as follows:

$$\begin{aligned} f(p \mid X) &= g(V_p \mid V_X) \\ &\geq g(V_p \mid V_Y) \\ &= f(p \mid V_Y), \end{aligned}$$

where the inequality comes from $V_X \subseteq V_Y$ and the submodularity of g; it may occur that V_p is included in V_Y (and V_X), but in such a case we have $f(p \mid Y) = 0$ (and $f(p \mid X) = 0$), which does not affect the conclusion. The monotonicity of f is confirmed readily from the monotonicity of g, and $f(\emptyset) = 0$ comes from $g(\emptyset) = 0$.

We then see that c is a submodular function. For $X \subseteq Y$ and $p \notin Y$, the diminishing return property holds as follows:

$$\begin{aligned} c(p \mid X) &= \sum_{v \in V_p \setminus V_X} \ell_v \\ &\geq \sum_{v \in V_p \setminus V_Y} \ell_v \\ &= c(p \mid Y), \end{aligned}$$

where we use $V_p \setminus V_Y \subseteq V_p \setminus V_X$ and $\ell_v \geq 0$ ($v \in V$). Similar to the above, $V_p \subseteq V_Y$ (and $V_p \subseteq V_X$) does not affect the conclusion. The monotonicity of c and $c(\emptyset) = 0$ are also easily obtained.

B Proof of Theorem 1

As is customary in the analysis of greedy algorithms for submodular knapsack problems (Khuller et al., 1999; Sviridenko, 2004), we introduce the following indexing of selected elements in $\mathcal{P}$. Let $X^* \subseteq \mathcal{P}$ be an optimal solution and t be the number of iterations executed by the algorithm until the first time at which $p \in X^*$ is considered but not added to the output solution, X, because of the violation of the knapsack constraint. We denote the number of elements added in the first t steps by d. If $c(X + p) > L$ and $p \notin X^*$ occur in the loops of the algorithm, then such p does not affect the analysis of approximation ratio. Therefore, we suppose that such p is removed from $\mathcal{P}$ in advance. Considering the above, we can define a sequence $p_1, p_2, \dots$ so that p_i is the i-th element added to X for $i = 1, \dots, d$ and p_{d+1} is the first element in X^* that is considered by the algorithm but not added to X due to the violation of the knapsack constraint. We define $X_i \coloneqq \{p_1, \dots, p_i\}$ for $i = 1, \dots, d+1$ and $X_0 \coloneqq \emptyset$.

For given subset $Q = \{q_1, \dots, q_K\} \subseteq \mathcal{P}$, path $\hat{q} \in Q$ is said to be *maximal* in Q if no $q \in Q$ satisfies $V_{\hat{q}} \subsetneq V_q$. A set of paths, $\hat{Q} \subseteq Q$, is a *maximal path cover* (MPC) of Q if all $\hat{q} \in \hat{Q}$ are maximal in Q and $V_{\hat{Q}} = V_Q$ holds. Since Q is defined on tree $\mathbf{T}$, any $Q \subseteq \mathcal{P}$ has a unique MPC $\hat{Q} \subseteq \mathcal{P}$. Furthermore, for any $q \in Q$, there exists at least one $\hat{q} \in \hat{Q}$ satisfying $V_q \subseteq V_{\hat{q}}$.

Lemma 1. *Given any $Z, Z^* \subseteq \mathcal{P}$, we define $\{q_1, \dots, q_K\} \coloneqq Z^* - Z$, $Z_j \coloneqq Z + \{q_1, \dots, q_j\}$ ($j \in [K]$) and $Z_0 \coloneqq Z$. Then the MPC $\{\hat{q}_1, \dots, \hat{q}_M\}$ of $Z^* - Z$ satisfies*

$$\sum_{j=1}^{K} f(q_j \mid Z_{j-1}) = \sum_{j=1}^{M} f(\hat{q}_j \mid \hat{Z}_{j-1}),$$

where $\hat{Z}_j \coloneqq Z + \{\hat{q}_1, \dots, \hat{q}_j\}$ and $\hat{Z}_0 \coloneqq Z$.

Proof. Since $\{\hat{q}_1, \dots, \hat{q}_M\}$ is the MPC of $Z^* - Z$, for any $q \in Z^* - Z$, there exists a $\hat{q} \in \{\hat{q}_1, \dots, \hat{q}_M\}$ satisfying $V_q \subseteq V_{\hat{q}}$. Therefore, $Z^* - Z$ can be divided into M subsets $\{q^i_1, \dots, q^i_{k_i}\}$ ($i \in [M]$) satisfying

$$V_{q^i_1} \subseteq \cdots \subseteq V_{q^i_{k_i}} = V_{\hat{q}_i}. \tag{A1}$$

Namely, $q^i_1, \dots, q^i_{k_i}$ are subpaths of $\hat{q}_i$; if some $q \in Q$ is included in multiple maximal paths, we arbitrarily choose one such maximal path to which q belongs. Thus all elements in $Z^* - Z$ are indexed as follows:

$$\begin{aligned} &Z^* - Z \\ &= \{q^1_1, \dots, q^1_{k_1}, q^2_1, \dots, q^2_{k_2}, \dots, q^M_1, \dots, q^M_{k_M}\}. \end{aligned}$$

We define $q^i_{j:k} \coloneqq \{q^i_j, q^i_{j+1}, \dots, q^i_k\}$ if $j \leq k$ and $q^i_{j:k} \coloneqq \emptyset$ otherwise. For any maximal path $\hat{q}_i \in \{\hat{q}_1, \dots, \hat{q}_M\}$ and any $\hat{Z}$ such that $Z \subseteq \hat{Z} \subseteq Z^*$, we have

$$
\begin{aligned}
&f(\hat{q}_i \mid \hat{Z}) \\
&= g(V_{\hat{Z}} \cup V_{\hat{q}_i}) - g(V_{\hat{Z}}) \\
&= g(V_{\hat{Z}} \cup V_{q^i_{k_i}}) - g(V_{\hat{Z}} \cup V_{q^i_{k_i-1}}) \\
&\quad + g(V_{\hat{Z}} \cup V_{q^i_{k_i-1}}) - g(V_{\hat{Z}} \cup V_{q^i_{k_i-2}}) \\
&\quad + \cdots \\
&\quad + g(V_{\hat{Z}} \cup V_{q^i_1}) - g(V_{\hat{Z}}) \\
&= g(V_{\hat{Z}} \cup V_{q^i_{1:k_i}}) - g(V_{\hat{Z}} \cup V_{q^i_{1:k_i-1}}) \\
&\quad + g(V_{\hat{Z}} \cup V_{q^i_{1:k_i-1}}) - g(V_{\hat{Z}} \cup V_{q^i_{1:k_i-2}}) \\
&\quad + \cdots \\
&\quad + g(V_{\hat{Z}} \cup V_{q^i_1}) - g(V_{\hat{Z}}) \\
&= f(q^i_{k_i} \mid \hat{Z} + q^i_{1:k_i-1}) + f(q^i_{k_i-1} \mid \hat{Z} + q^i_{1:k_i-2}) \\
&\quad + \cdots + f(q^i_1 \mid \hat{Z}),
\end{aligned}
$$

where the third equality comes from (A1). Note that the value of $\sum_{j\in[K]} f(q_j \mid Z_{j-1}) = f(Z^*) - f(Z)$ is independent of the order of elements in $Z^* - Z$. Thus, rearranging the order of summation yields

$$
\begin{aligned}
\sum_{j=1}^{K} f(q_j \mid Z_{j-1}) &= \sum_{i=1}^{M}\sum_{j=1}^{k_i} f(q^i_j \mid \hat{Z}_{i-1} + q^i_{1:j-1}) \\
&= \sum_{j=1}^{M} f(\hat{q}_j \mid \hat{Z}_{j-1}).
\end{aligned}
$$

∎

For an optimal subtree $X^* \subseteq \mathcal{P}$ in $\mathbf{T}$, we let X^*_i denote a subtree of X^* that is included in the i-th sentence tree T_i ($i \in [N]$). We define λ_i as the number of leaves of T_i. Note that, if $Q_i \subseteq \mathcal{P}$ is the MPC of X^*_i, then we have $|Q_i| \le \lambda_i$ (i.e., the number of paths in MPC is bounded by the number of leaves). Let $\lambda := \max_{i\in[N]} \lambda_i$. Then we have the following lemma.

Lemma 2. *For $i = 1, \ldots, d+1$, we have*

$$
\begin{aligned}
&f(X_i) - f(X_{i-1}) \\
&\ge \frac{c(p_i \mid X_{i-1})}{\lambda L}(f(X^*) - f(X_{i-1})).
\end{aligned}
$$

Proof. Let $\{q_1, \ldots, q_K\} := X^* - X_{i-1}$, $Z_j := X_{i-1} + \{q_1, \ldots, q_j\}$ and $Z_0 := X_{i-1}$. From Lemma 1 with $Z^* = X^*$ and $Z = X_{i-1}$, MPC $\hat{Q} = \{\hat{q}_1, \ldots, \hat{q}_M\}$ of $X^* - X_{i-1}$ satisfies

$$
\begin{aligned}
f(X^*) - f(X_{i-1}) &= \sum_{j=1}^{K} f(q_j \mid Z_{j-1}) \\
&= \sum_{j=1}^{M} f(\hat{q}_j \mid \hat{Z}_{j-1}),
\end{aligned}
$$

where $\hat{Z}_j := X_{i-1} + \{\hat{q}_1, \ldots, \hat{q}_j\}$ ($j \in [M]$) and $\hat{Z}_0 = X_{i-1}$. By using submodularity, we obtain

$$
\begin{aligned}
f(X^*) - f(X_{i-1}) &= \sum_{j=1}^{M} f(\hat{q}_j \mid \hat{Z}_{j-1}) \\
&\le \sum_{j=1}^{M} f(\hat{q}_j \mid \hat{Z}_0) \\
&= \sum_{j=1}^{M} f(\hat{q}_j \mid X_{i-1}).
\end{aligned}
$$

Since $p_i = \mathrm{argmax}_{p\notin X_{i-1}} \frac{f(p|X_{i-1})}{c(p|X_{i-1})}$ holds, we have $\frac{f(p_i|X_{i-1})}{c(p_i|X_{i-1})} \ge \frac{f(\hat{q}_j|X_{i-1})}{c(\hat{q}_j|X_{i-1})}$ for all $j = 1, \ldots, M$. Hence we obtain

$$
\begin{aligned}
&c(p_i \mid X_{i-1})(f(X^*) - f(X_{i-1})) \qquad \text{(A2)} \\
&\le c(p_i \mid X_{i-1}) \sum_{j=1}^{M} f(\hat{q}_j \mid X_{i-1}) \\
&\le f(p_i \mid X_{i-1}) \sum_{j=1}^{M} c(\hat{q}_j \mid X_{i-1}).
\end{aligned}
$$

We now bound $\sum_{j=1}^{M} c(\hat{q}_j \mid X_{i-1})$ from above as follows. By using submodularity, we obtain

$$
\sum_{j=1}^{M} c(\hat{q}_j \mid X_{i-1}) \le \sum_{j=1}^{M} c(\hat{q}_j). \qquad \text{(A3)}
$$

Note that $\hat{Q} = \{\hat{q}_1, \ldots, \hat{q}_M\}$ can be partitioned into N subsets $Q_1, \ldots, Q_N$ of maximal paths so that all $q \in Q_i$ include r_i; we have $V_{Q_i} \cap V_{Q_j} = \emptyset$ for $i \ne j$ since each Q_i ($i \in [N]$) is defined on the i-th sentence tree, T_i. Using these definitions, we obtain

$$
\sum_{j=1}^{M} c(\hat{q}_j) = \sum_{i\in[N]}\sum_{q\in Q_i} c(q) = \sum_{i\in[N]}\sum_{q\in Q_i}\sum_{v\in V_q} \ell_v.
$$

Since we have $|Q_i| \le \lambda_i$, each $v \in V_{Q_i}$ is included in at most λ_i maximal paths in Q_i. Thus we have

$$
\sum_{q\in Q_i}\sum_{v\in V_q} \ell_v \le \lambda_i \sum_{v\in V_{Q_i}} \ell_v \le \lambda \sum_{v\in V_{Q_i}} \ell_v.
$$

Furthermore, since $\hat{Q} = \{\hat{q}_1, \dots, \hat{q}_M\} \subseteq X^*$ satisfies the knapsack constraint, we have

$$\sum_{i\in[N]} \sum_{v\in V_{Q_i}} \ell_v = \sum_{v\in V_{\hat{Q}}} \ell_v = c(\{\hat{q}_1, \dots, \hat{q}_M\}) \le L.$$

From the above inequalities, we obtain

$$\sum_{j=1}^{M} c(\hat{q}_j) = \sum_{i\in[N]} \sum_{q\in Q_i} \sum_{v\in V_q} \ell_v \le \lambda \sum_{i\in[N]} \sum_{v\in V_{Q_i}} \ell_v \le \lambda L. \tag{A4}$$

Combining (A2), (A3) and (A4), we obtain

$$c(p_i \mid X_{i-1})(f(X^*) - f(X_{i-1})) \le f(p_i \mid X_{i-1})\lambda L.$$

The claim follows by rearranging terms and using $f(p_i \mid X_{i-1}) = f(X_i) - f(X_{i-1})$. ∎

Lemma 3. *For $i = 1, \dots, d+1$, we have*

$$f(X_i) \ge \left(1 - \prod_{k=1}^{i} \left(1 - \frac{c(p_k \mid X_{k-1})}{\lambda L}\right)\right) f(X^*).$$

Proof. We prove the lemma by induction on $i = 1, \dots, d+1$. First, if $i = 1$, we have $X_1 = \{p_1\}$ and thus the claim follows by Lemma 2. Then we assume the lemma holds for $X_1, \dots, X_{i-1}$ and prove that it holds for X_i. Combining Lemma 2 and the assumption, we obtain

$$\begin{aligned} f(X_i) &= f(X_{i-1}) + (f(X_i) - f(X_{i-1})) \\ &\ge f(X_{i-1}) + \frac{c(p_i \mid X_{i-1})}{\lambda L}(f(X^*) - f(X_{i-1})) \\ &= \left(1 - \frac{c(p_i \mid X_{i-1})}{\lambda L}\right) f(X_{i-1}) \\ &\quad + \frac{c(p_i \mid X_{i-1})}{\lambda L} f(X^*) \\ &\ge \left(1 - \prod_{k=1}^{i} \left(1 - \frac{c(p_k \mid X_{k-1})}{\lambda L}\right)\right) f(X^*). \end{aligned}$$

Thus the lemma holds by induction. ∎

Theorem 1. *Algorithm 1 achieves at least $\frac{1}{2}(1 - e^{-1/\lambda})$-approximation.*

Proof. Since $\sum_{k=1}^{d+1} \frac{c(p_k|X_{k-1})}{c(X_{d+1})} = 1$ holds, $\prod_{k=1}^{d+1} \left(1 - \frac{1}{\lambda} \cdot \frac{c(p_k|X_{k-1})}{c(X_{d+1})}\right)$ attains its maximum when we have $\frac{c(p_1|X_0)}{c(X_{d+1})} = \cdots = \frac{c(p_{d+1}|X_d)}{c(X_{d+1})} = \frac{1}{d+1}$. Namely, the following inequality holds:

$$\prod_{k=1}^{d+1} \left(1 - \frac{1}{\lambda} \cdot \frac{c(p_k \mid X_{k-1})}{c(X_{d+1})}\right) \le \left(1 - \frac{1}{\lambda} \cdot \frac{1}{d+1}\right)^{d+1}.$$

By using Lemma 3, the above inequality, and the fact that the knapsack constraint is violated by adding $(d+1)$-th element (i.e., $c(X_{d+1}) > L$), we obtain

$$\begin{aligned} f(X_{d+1}) &\ge \left(1 - \prod_{k=1}^{d+1} \left(1 - \frac{c(p_k \mid X_{k-1})}{\lambda L}\right)\right) f(X^*) \\ &\ge \left(1 - \prod_{k=1}^{d+1} \left(1 - \frac{1}{\lambda} \cdot \frac{c(p_k \mid X_{k-1})}{c(X_{d+1})}\right)\right) f(X^*) \\ &\ge \left(1 - \left(1 - \frac{1}{\lambda} \cdot \frac{1}{d+1}\right)^{d+1}\right) f(X^*) \\ &\ge \left(1 - \frac{1}{e^{1/\lambda}}\right) f(X^*). \end{aligned}$$

This leads to the following inequality:

$$\begin{aligned} f(X_{d+1}) &= f(X_d) + f(p_{d+1} \mid X_d) \\ &\ge (1 - e^{-1/\lambda}) f(X^*). \end{aligned}$$

We note that the solution, X, obtained by Steps 1–8 in Algorithm 1 satisfies $f(X) \ge f(X_d)$ and that $\hat{p}$ chosen in Step 9 satisfies $f(\hat{p}) \ge f(p_{d+1} \mid X_d)$. Therefore, the output of Algorithm 1, which is defined as $Y \coloneqq \operatorname{argmax}_{X' \in \{X, \hat{p}\}} f(X')$, satisfies $f(Y) \ge \frac{1}{2}(1 - e^{-1/\lambda}) f(X^*)$. ∎

C ILP formulations

We present ILP formulations for the three objective functions described in Section 5. In the experiments, the ILP-based method obtained summaries by solving the following optimization problems.

Coverage Function

The ILP formulation with the coverage function can be written as follows:

$$\underset{z,b}{\text{maximize}} \quad \sum_{j=1}^{M} w_j z_j \tag{A5}$$
$$\text{subject to} \quad \sum_{v \in V} \ell_v b_v \le L, \tag{A6}$$
$$\forall v \in V \backslash r_{1:N}: \quad b_{\text{parent}(v)} \ge b_v, \tag{A7}$$
$$\forall j \in [M]: \quad \sum_{v \in V_j} b_v \ge z_j, \tag{A8}$$
$$\forall v \in V: \quad b_v \in \{0,1\},$$
$$\forall j \in [M]: \quad z_j \in \{0,1\}.$$

z_j is a binary decision variable that indicates whether the j-th word is contained in the summary or not. b_v is a binary decision variable that represents whether chunk $v \in V$ is contained in the summary or not.

Constraint (A6) guarantees that the obtained summary includes at most L words. Remember that $r_i \in V$ ($i \in [N]$) is the root node of dependency tree T_i constructed for the i-th sentence; we use $r_{1:N}$ as shorthand for $\{r_1, \ldots, r_N\}$. Function parent(v) returns the parent chunk of $v \in V$ in the dependency trees. Therefore, constraint (A7) guarantees that the obtained summary comprises some rooted subtrees of the dependency trees. $V_j \subseteq V$ denotes the set of all chunks that include the j-th word. Thus, constraint (A8) means that at least one chunk including the j-th word must be chosen in order to cover the j-th word.

Coverage Function with Rewords

The ILP formulation for this objective function can be obtained by replacing the objective function in (A5) with

$$\sum_{j=1}^{M} w_j z_j - \gamma \left(\sum_{v \in V} \ell_v b_v - \sum_{i=1}^{N} b_{r_i} \right),$$

where γ is a hyper parameter that balances the total weight of covered chunks and the positive reword term.

ROUGE$_1$

As in (Hirao et al., 2017), compressive summarization with the ROUGE$_1$ objective function can be formulated as the following ILP:

$$\underset{z,b}{\text{maximize}} \quad \sum_{k=1}^{K} \sum_{j=1}^{M} z_{k,j}$$
$$\text{subject to} \quad \sum_{v \in V} \ell_v b_v \le L,$$
$$\forall k \in [K], j \in [M]: \quad \mathrm{C}_{e_j}(R_k) \ge z_{k,j}, \tag{A9}$$
$$\forall k \in [K], j \in [M]: \quad \sum_{v \in V_j} b_v \ge z_{k,j}, \tag{A10}$$
$$\forall v \in V \backslash r_{1:N}: \quad b_{\text{parent}(v)} \ge b_v,$$
$$\forall v \in V: \quad b_v \in \{0,1\},$$
$$\forall k \in [K], j \in [M]: \quad z_{k,j} \in \mathbb{Z}_{\ge 0}.$$

We here suppose that the document data contains M distinct unigrams indexed with $j \in [M]$; e_j denotes the j-th unigram, and $V_j \subseteq V$ is the set of all chunks that include e_j. Each non-negative integer variable $z_{k,j}$ counts the number of times that e_j appears both in the k-th reference summary and in the summary to be output, which we denote by $S \subseteq V$. From constraints (A9), (A10), and $\sum_{v \in V_j} b_v = \mathrm{C}_{e_j}(S)$, we see that the objective function corresponds to the numerator of ROUGE (3) with $n = 1$. The remaining parts are similar to those in the ILP formulation for the coverage function.

References

Tsutomu Hirao, Masaaki Nishino, and Masaaki Nagata. 2017. Oracle summaries of compressive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 275–280. https://doi.org/10.18653/v1/P17-2043.

Samir Khuller, Anna Moss, and Joseph S. Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters* 70(1):39–45. https://doi.org/10.1016/S0020-0190(99)00031-9.

Maxim Sviridenko. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* 32(1):41–43. https://doi.org/10.1016/S0167-6377(03)00062-2.