Single-Document Summarization as a Tree Knapsack Problem

Tsutomu Hirao[†] **Yasuhisa Yoshida**[†] **Masaaki Nishino**[†] **Norihito Yasuda**[‡] **Masaaki Nagata**[†]
†NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{hirao.tsutomu,yoshida.y,nishino.masaaki,
nagata.masaaki}@lab.ntt.co.jp
‡ Japan Science and Technology Agency
North 14 West 9, Sapporo, Hokkaido, 060-0814, Japan
yasuda@erato.ist.hokudai.ac.jp

Abstract

Recent studies on extractive text summarization formulate it as a combinatorial optimization problem such as a *Knapsack Problem*, a *Maximum Coverage Problem* or a *Budgeted Median Problem*. These methods successfully improved summarization quality, but they did not consider the rhetorical relations between the textual units of a source document. Thus, summaries generated by these methods may lack logical coherence. This paper proposes a single document summarization method based on the trimming of a discourse tree. This is a two-fold process. First, we propose rules for transforming a rhetorical structure theory-based discourse tree into a dependency-based discourse tree, which allows us to take a tree-trimming approach to summarization. Second, we formulate the problem of trimming a dependency-based discourse tree as a *Tree Knapsack Problem*, then solve it with integer linear programming (ILP). Evaluation results showed that our method improved ROUGE scores.

1 Introduction

State-of-the-art extractive text summarization methods regard a document (or a document set) as a set of textual units (e.g. sentences, clauses, phrases) and formulate summarization as a combinatorial optimization problem, *i.e.* selecting a subset of the set of textual units that maximizes an objective without violating a length constraint. For example, McDonald (2007) formulated text summarization as a *Knapsack Problem*, where he selects a set of textual units that maximize the sum of significance scores of each unit. Filatova et al. (2004) proposed a summarization method based on a *Maximum Coverage Problem*, in which they select a set of textual units that maximizes the weighted sum of the conceptual units (e.g. unigrams) contained in the set. Although, their greedy solution is only an approximation, Takamura et al. (2009a) extended it to obtain the exact solution. More recently, Takamura et al. (2009b) regarded summarization as a *Budgeted Median Problem* and obtain exact solutions with integer linear programming.

These methods successfully improved ROUGE (Lin, 2004) scores, but they still have one critical shortcoming. Since these methods are based on subset selection, the summaries they generate cannot preserve the rhetorical structure of the textual units of a source document. Thus, the resulting summary may lack coherence and may not include significant textual units from a source document.

One powerful and potential way to overcome the problem is to include discourse tree constraints in the summarization procedure. Marcu (1998) regarded a document as a Rhetorical Structure Theory (RST) (William Charles, Mann and Sandra Annear, Thompson, 1988)-based discourse tree (RST-DT) and selected textual units according to a preference ranking derived from the tree structure to make a summary. Daumé et al. (2002) proposed a document compression method that directly models the probability of a summary given an RST-DT by using a noisy-channel model. These methods generate well-organized summaries, however, since they do not formulate summarizations as combinatorial op-

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1515–1520,
Seattle, Washington, USA, 18-21 October 2013.

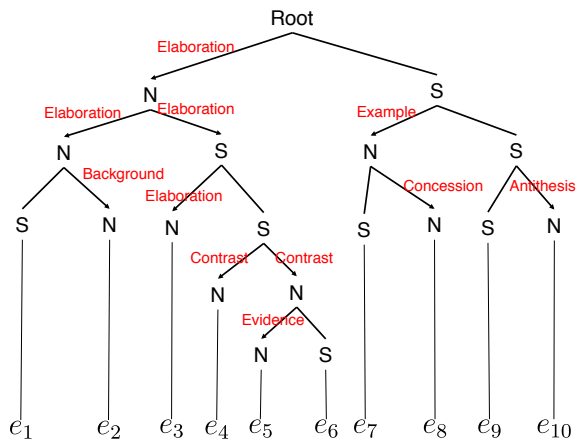

Figure 1: Example RST-DT from (Marcu, 1998).

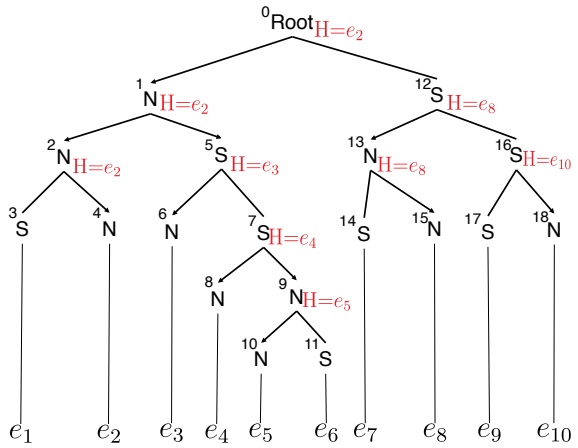

Figure 2: Heads of non-terminal nodes.

timization problems, the optimality of the generated summaries is not guaranteed.

In this paper, we propose a single document summarization method based on the trimming of a discourse tree based on the *Tree Knapsack Problem*. If a discourse tree explicitly represents parent-child relationships between textual units, we can apply the well-known tree-trimming approach to a discourse tree and reap the benefit of combinatorial optimization methods. In other words, to apply the tree-trimming approach, we need a tree whose all nodes represent textual units. Unfortunately, the RST-DT does not allow it, because textual units in the RST-DT are located only on leaf nodes and parent-child relationship between textual units are represented implicitly at higher positions in a tree. Therefore, we first propose rules that transform an RST-DT into a dependency-based discourse tree (DEP-DT) that explicitly defines the parent-child relationships. Second, we treat it as a rooted subtree selection, in other words, a *Tree Knapsack Problem* and formulate the problem as an ILP.

2 From RST-DT to DEP-DT

2.1 RST-DT

According to RST, a document is represented as an RST-DT whose terminal nodes correspond to elementary discourse units (EDU)s[1] and whose non-terminal nodes indicate the role of the contiguous EDUs namely, 'nucleus (N)' or 'satellite (S)'. A nucleus is more important than a satellite in terms of the writer's purpose. That is, a satellite is a child of a nucleus in the RST-DT. Some discourse relations such as 'Elaboration', 'Contrast' and 'Evidence' between a nucleus and a satellite or two nuclei are defined. Figure 1 shows an example of an RST-DT.

2.2 DEP-DT

An RST-DT is not suitable for tree trimming because it does not always explicitly define parent-child relationships between textual units. For example, if we consider how to trim the RST-DT in Figure 1, when we drop e_8, we have to drop e_7 because of the parent-child relationship defined between e_7 and e_8, *i.e.* e_7 is a satellite (child) of the nucleus (parent) e_8. On the other hand, we cannot judge whether we have to drop e_9 or e_{10} because the parent-child relationships are not explicitly defined between e_8 and e_9, e_8 and e_{10}. This view motivates us to produce a discourse tree that explicitly defines parent-child relationships and whose root node represents the most important EDU in a source document. If we can obtain such a tree, it is easy to formulate summarization as a *Tree Knapsack Problem*.

To construct a discourse tree that represents the parent-child relationships between EDUs, we propose rules for transforming an RST-DT to a dependency-based discourse tree (DEP-DT). The procedure is defined as follows:

1. For each non-terminal node excluding the par-

[1] EDUs roughly correspond to clauses.

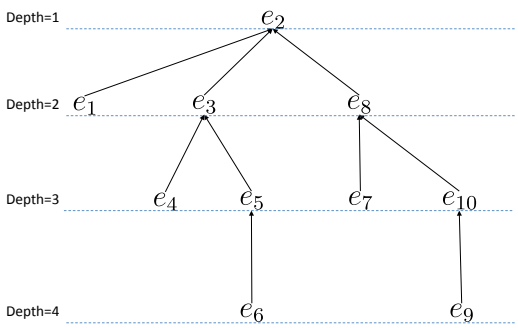

Figure 3: The DEP-DT obtained from the RST-DT in Figure 1.

ent of an EDU in the RST-DT, we define a 'head'. Here, a 'head' of a non-terminal node is the leftmost descendant EDU whose parent is N. In Figure 2, 'H' indicates the 'head' of each node.

2. For each EDU whose parent is N, we pick the nearest S with a 'head' from the EDU's ancestors and we add the EDU to the DEP-DT as a child of the head of the S's parent. If there is no nearest S, the EDU is the root of the DEP-DT. For example, in Figure 2, the nearest S to e_3 that has a head is node 5 and the head of node 5's parent is e_2. Thus, e_3 is a child of e_2.

3. For each EDU whose parent is S, we pick the nearest non-terminal with a 'head' from the ancestors and we add the EDU to the DEP-DT as a child of the head of the non-terminal node. For example, the nearest non-terminal node of e_9 that has a head is node 16 and the head of node 16 is e_{10}. Thus, e_9 is a child of e_{10}.

Figure 3 shows the DEP-DT obtained from the RST-DT in Figure 1. The DEP-DT expresses the parent-child relationship between the EDUs. Therefore, we have to drop e_7, e_9 and e_{10} when we drop e_8. Note that, by applying the rules, discourse relations defined between non-terminals of an RST-DT are eliminated. However, we believe that these relations are no needed for the summarization that we are attempting to realize.

3 Tree Knapsack Model for Single-Document Summarization

3.1 Formalization

We denote T as a set of all possible rooted subtrees obtained from a DEP-DT. $F(t)$ is the significance score for a rooted subtree $t \in T$ and L is the maximum number of words allowed in a summary. The optimal subtree t^* is defined as follows:

$$t^* = \arg\max_{t \in T} F(t) \quad (1)$$

$$s.t. \quad \text{Length}(t) \leq L. \quad (2)$$

Here, we define $F(t)$ as

$$F(t) = \sum_{e \in E(t)} \frac{\mathcal{W}(e)}{\text{Depth}(e)}. \quad (3)$$

$E(t)$ is the set of EDUs contained in t, $\text{Depth}(e)$ is the depth of an EDU e within the DEP-DT. For example, $\text{Depth}(e_2) = 1$, $\text{Depth}(e_6) = 4$ for the DEP-DT of Figure 3. $\mathcal{W}(e)$ is defined as follows:

$$\mathcal{W}(e) = \sum_{w \in W(e)} \text{tf}(w, D). \quad (4)$$

$W(e)$ is the set of words contained in e and $\text{tf}(w, D)$ is the term frequency of word w in a document D.

3.2 ILP Formulation

We formulate the optimization problem in the previous section as a *Tree Knapsack Problem*, which is a kind of *Precedence-Constrained Knapsack Problem* (Samphaiboon and Yamada, 2000) and we can obtain the optimal rooted subtree by solving the following ILP problem[2]:

$$\underset{\boldsymbol{x}}{\text{maximize}} \quad \sum_{i=1}^{N} \frac{\mathcal{W}(e_i)}{\text{Depth}(e_i)} x_i \quad (5)$$

$$s.t. \quad \sum_{i=1}^{N} \ell_i x_i \leq L \quad (6)$$

$$\forall i : x_{\text{parent}(i)} \geq x_i \quad (7)$$

$$\forall i : x_i \in \{0, 1\}, \quad (8)$$

[2]A similar approach has been applied to sentence compression (Filippova and Strube, 2008).

	ROUGE-1		ROUGE-2	
	F	R	F	R
TKP(G)	**.310**^{H,K,L}	**.321**^{G,H,K,L}	**.108**	**.112**^{H}
TKP(H)	.281^{H}	.284^{H}	.092	.093
Marcu(G)	.291^{H}	.272^{H}	.101	.093
Marcu(H)	.236	.219	.073	.068
MCP	.279	.295^{H}	.073	.077
KP	.251	.266^{H}	.071	.075
LEAD	.255	.240	.092	.086

Table 1: ROUGE scores of the RST discourse treebank dataset. In the table, ^{G,H,K,L} indicate a method statistically significant against Marcu(G), Marcu(H), KP, LEAD, respectively.

where $\boldsymbol{x}$ is an N-dimensional binary vector that represents the summary, *i.e.* x_i=1 denotes that the i-th EDU is included in the summary. N is the number of EDUs in a document, ℓ_i is the length (the number of words) of the i-th EDU, and parent(i) indicates the ID of the parent of the i-th EDU in the DEP-DT. Constraint (6) ensures that the length of a summary is less than limit L. Constraint (7) ensures that a summary is a rooted subtree of the DEP-DT. Thus, $x_{\text{parent}(i)}$ is always 1 when the i-th EDU is included in the summary.

In general, the *Tree Knapsack Problem* is NP-hard, but fortunately we can obtain the optimal solution in a feasible time by using ILP solvers for documents of practical tree size. In addition, bottom-up DP (Lukes, 1974) and depth-first DP algorithms (Cho and Shaw, 1997) are known to find the optimal solution efficiently.

4 Experimental Evaluation

4.1 Settings

We conducted an experimental evaluation on the test collection for single document summarization evaluation contained in the RST Discourse Treebank (RST-DTB)(Carlson et al., 2001) distributed by the Linguistic Data Consortium (LDC)[3]. The RST-DTB Corpus includes 385 Wall Street Journal articles with RST annotation, and 30 of these documents also have one human-made reference summary. The average length of the reference summaries corresponds to about 10 % of the words in the source document.

We compared our method (TKP) with Marcu's method (Marcu) (Marcu, 1998), a simple knapsack model (KP), a maximum coverage model (MCP) and a lead method (LEAD). MCP is known to be a state-of-the-art method for multiple document summarization and we believe that MCP also performs well in terms of single document summarization. LEAD is also a widely used summarizer that simply takes the first K textual units of the document. Although this is a simple heuristic rule, it is known as a state-of-the-art summarizer (Nenkova and McKeown, 2011).

For our method, we examined two types of DEP-DT. One was obtained from the gold RST-DT. The other was obtained from the RST-DT produced by a state-of-the-art RST parser, HILDA (duVerle and Prendinger, 2009; Hernault et al., 2010). For Marcu's method, we examined both the gold RST-DT and HILDA's RST-DT. We re-implemented HILDA and re-trained it on the RST-DT Corpus excluding the 30 documents used in the evaluation. The F-score of the parser was around 0.5. For KP, we exclude constraint (7) from the ILP formulation of TKP and set the depth of all EDUs in equations (3) and (5) at 1. For MCP, we use tf (equation (4)) as the word weight.

We evaluated the summarization systems with ROUGE version 1.5.5 [4]. Performance metrics were the recall (R) and F-score (F) of ROUGE-1,2.

4.2 Results and Discussion

Table 1 shows the evaluation results. In the table, TKP(G) and TKP(H) denote methods with the DEP-DT obtained from the gold RST-DT and from HILDA, respectively. Marcu(G) and Marcu(H) denote Marcu's method described in (Marcu, 1998) with gold RST-DT and with HILDA, respectively. We performed a multiple comparison test for the differences among ROUGE scores, we calculated the p-values between systems with the Wilcoxon signed-rank test (Wilcoxon, 1945) and used the False Discovery Rate (FDR) (Benjamini and Hochberg, 1995) to calculate adjusted p-values, in order to limit false positive rate to 5%.

From the table, TKP(G) and Marcu(G) achieved

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07

[4] Options used: -n 2 -s -m -x

Reference:
The Fuji apple may one day replace the Red Delicious as the number one U.S. apple. Since the Red Delicious has been over-planted and prices have dropped to new lows, the apple industry seems ready for change. Along with growers, supermarkets are also trying different varieties of apples. Although the Fuji is smaller and not as perfectly shaped as the Red Delicious, it is much sweeter, less mealy and has a longer shelf life.

TKP(G):
We'll still have mom and apple pie. A Japanese apple called the Fuji. Some fruit visionaries say the Fuji could someday tumble the Red Delicious from the top of America's apple heap. It has a long shelf life. Now, even more radical changes seem afoot. The Delicious hegemony won't end anytime soon. New apple trees grow slowly. But the apple industry is ripe for change. There's a Fuji apple cult.

Marcu(G):
We'll still have mom and apple pie. On second thought, make that just mom. The Fuji could someday tumble the Red Delicious from the top of America's apple heap. Now, even more radical changes seem afoot. The Delicious hegemony won't end anytime soon. More than twice as many Red Delicious apples are grown as the Golden variety, America's No. 2 apple. But the apple industry is ripe for change.

MCP:
Called the Fuji. It has a long shelf life. New apple trees grow slowly. Its roots are patriotic. I'm going to have to get another job this year. Scowls. They still buy apples mainly for big, red good looks. Japanese researchers have bred dozens of strains of Fujis. Mr. Auvil, the Washington grower, says. Stores sell in summer. The " big boss " at a supermarket chain even rejected his Red Delicious recently. Many growers employ.

LEAD:
Soichiro Honda's picture now hangs with Henry Ford's in the U.S. Automotive Hall of Fame, and the game-show " Jeopardy " is soon to be Sony-owned. But no matter how much Japan gets under our skin, we'll still have mom and apple pie. On second thought, make that just mom. A Japanese apple called the Fuji is cropping up in orchards the way Hondas did on U.S. roads.

Figure 4: Summaries obtained from wsj_1128.

better results than MCP, KP and LEAD, although some of the comparisons are not significant. In particular, TKP(G) achieved the highest ROUGE scores on all measures. On ROUGE-1 Recall, TKP(G) significantly outperformed Marcu(G), Marcu(H), KP and LEAD. These results support the effectiveness of our method that utilizes the discourse structure. Comparing TKP(H) with Marcu(H), the former achieved higher scores with statistical significance on ROUGE-1. In addition, Marcu(H) was outperformed by MCP, KP and LEAD. The results confirm the effectiveness of our summarization model and trimming proposal for DEP-DT. Moreover, the difference between TKP(G) and TKP(H) was smaller than that between Marcu(G) and Marcu(H). This implies that our method is more robust against discourse parser error than Marcu's method.

Figure 4 shows the example summaries generated by TKP(G), Marcu(G), MCP and LEAD, respectively for an article, wsj_1128. Since TKP(G) and Marcu(G) utilize a discourse tree, the summary generated by TKP(G) is similar to that generated by Marcu(G) but it is different from those generated by MCP and LEAD.

5 Conclusion

This paper proposed rules for transforming an RST-DT to a DEP-DT to obtain the parent-child relationships between EDUs. We treated a single document summarization method as a *Tree Knapsack Problem*, *i.e.* the summarizer selects the best rooted subtree from a DEP-DT. To demonstrate the effectiveness of our method, we conducted an experimental evaluation using 30 documents selected from the RST Discourse Treebank Corpus. The results showed that our method achieved the highest ROUGE-1,2 scores.

References

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, 57(1):289–300.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proc. of the SIGDIAL01*, pages 1–10.

Geon Cho and Dong X Shaw. 1997. A depth-first dynamic programming algorithm for the tree knap-

sack problem. *INFORMS Journal on Computing*, 9(4):431–438.

Hal Daumé III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proc. of the 40th ACL*, pages 449–456.

David duVerle and Helmut Prendinger. 2009. A novel discourse parser based on support vector machine classification. In *Proc. of the Joint Conference of the 47th ACL and 4th IJCNLP*, pages 665–673.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence extraction. In *Proc. of the 20th COLING*, pages 397–403.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proc. of the 5th International Natural Language Generation Conference (INLG)*, pages 25–32.

Hugo Hernault, Helmut Prendinger, David A duVerle, and Mitsuru Ishizuka. 2010. HILDA: A discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of Workshop on Text Summarization Branches Out*, pages 74–81.

J. A. Lukes. 1974. Efficient algorithm for the partitioning of trees. *IBM Journal of Research and Development*, 18(3):217–224.

Daniel Marcu. 1998. Improving summarization through rhetorical parsing tuning. In *Proc. of the 6th Workshop on Very Large Corpora*, pages 206–215.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proc. of the 29th ECIR*, pages 557–564.

Ani Nenkova and Kathaleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233.

Natthawut Samphaiboon and Takeo Yamada. 2000. Heuristic and exact algorithms for the precedence-constrained knapsack problem. *Journal of Optimization Theory and Applications*, 105(3):659–676.

Hiroya Takamura and Manabu Okumura. 2009a. Text summarization model based on maximum coverage problem and its variant. In *Proc. of the 12th EACL*, pages 781–789.

Hiroya Takamura and Manabu Okumura. 2009b. Text summarization model based on the budgeted median problem. In *Proceedings of the 18th CIKM*.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

William Charles, Mann and Sandra Annear, Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.