Dependency-based Discourse Parser for Single-Document Summarization

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
{yoshida.y,suzuki.jun,hirao.tsutomu,nagata.masaaki}@lab.ntt.co.jp

Abstract

The current state-of-the-art single-document summarization method generates a summary by solving a Tree Knapsack Problem (TKP), which is the problem of finding the optimal rooted subtree of the dependency-based discourse tree (DEP-DT) of a document. We can obtain a gold DEP-DT by transforming a gold Rhetorical Structure Theory-based discourse tree (RST-DT). However, there is still a large difference between the ROUGE scores of a system with a gold DEP-DT and a system with a DEP-DT obtained from an automatically parsed RST-DT. To improve the ROUGE score, we propose a novel discourse parser that directly generates the DEP-DT. The evaluation results showed that the TKP with our parser outperformed that with the state-of-the-art RST-DT parser, and achieved almost equivalent ROUGE scores to the TKP with the gold DEP-DT.

1 Introduction

Discourse structures of documents are believed to be highly beneficial for generating informative and coherent summaries. Several discourse-based summarization methods have been developed, such as (Marcu, 1998; Daumé III and Marcu, 2002; Hirao et al., 2013; Kikuchi et al., 2014). Moreover, the current best ROUGE score for the summarization benchmark data of the RST-discourse Treebank (Carlson et al., 2002) has been provided by (Hirao et al., 2013), whose method also utilizes discourse trees. Thus, the discourse-based summarization approach is one promising way to obtain high-quality summaries.

One possible weakness of discourse-based summarization techniques is that they rely greatly on the accuracy of the discourse parser they use. For example, the above discourse-based summarization methods utilize discourse trees based on the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) for their discourse information. Unfortunately, the current state-of-the-art RST parser, as described in (Hernault et al., 2010), is insufficient as an off-the-shelf discourse parser. In fact, there is empirical evidence that the quality (i.e., ROUGE score) of summaries from auto-parsed discourse trees is significantly degraded compared with those generated from gold discourse trees (Marcu, 1998; Hirao et al., 2013).

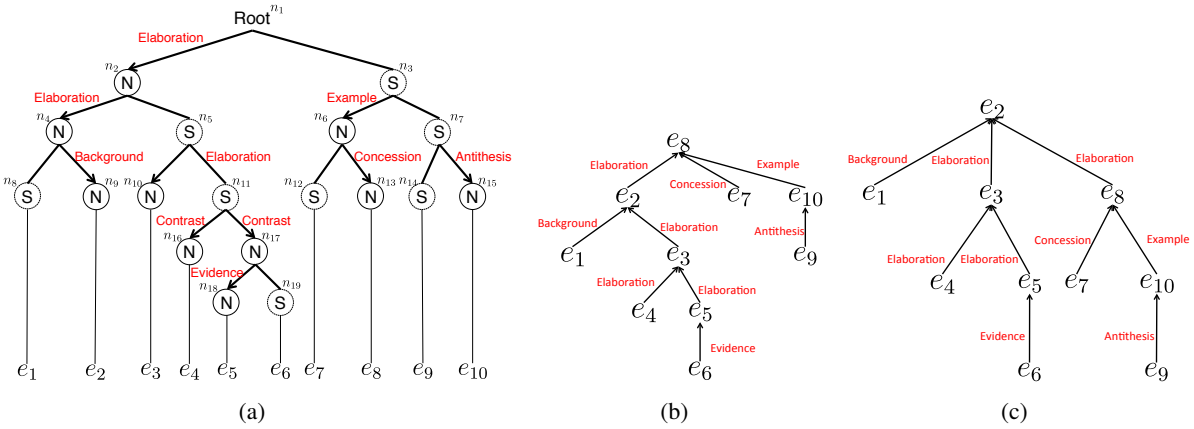
From this background, the goal of this paper is to develop an appropriate discourse parser for discourse-based summarization. We first focus on one of the best discourse-based single document summarization methods as proposed in (Hirao et al., 2013). Their method formulates a single document summarization problem as a Tree Knapsack Problem (TKP) over a dependency-based discourse tree (DEP-DT). In their method, DEP-DTs are automatically transformed from (auto-parsed) RST-discourse trees (RST-DTs) by heuristic rules. Instead, we develop a DEP-DT parser, that directly provides DEP-DTs for their state-of-the-art discourse-based summarization method. We show that summaries generated by our parser improve the ROUGE scores to almost the same level as those generated by gold DEP-DTs. We also investigate the way in which the parsing accuracy helps to improve the ROUGE scores.

2 Single-Document Summarization as a Tree Knapsack Problem

Hirao et al. (2013) formulated single-document summarization as a TKP that is run on the DEP-DT. They obtained a summary by trimming the DEP-DT, i.e. the summary is a rooted subtree of the DEP-DT.

Suppose that we have N EDUs in a document,

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1834–1839,
October 25-29, 2014, Doha, Qatar.



(a) (b) (c)

Figure 1: Examples of RST-DT and DEP-DT. $e_1, \cdots, e_{10}$ are EDUs. (a) Example of an RST-DT from (Marcu, 1998). $n_1, \cdots, n_{19}$ are the non-terminal nodes. (b) Example of the DEP-DT obtained from the incorrect RST-DT that is made by swapping the Nucleus-Satellite relationship of the node n_2 and the node n_3. (c) The correct DEP-DT obtained from the RST-DT in (a).

and the i-th EDU e_i has l_i words. L is the maximum number of words allowed in a summary. In the TKP, if we select e_i, we need to select its parent EDU in the DEP-DT. We denote parent(i) as the index of the parent of e_i in the DEP-DT. $\mathbf{x}$ is an N-dimensional binary vector that represents a summary, i.e. $x_i = 1$ denotes that e_i is included in the summary. The TKP is defined as the following ILP problem:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{N} F(e_i) x_i \\ s.t. \quad & \textstyle\sum_{i=1}^{N} l_i x_i \leq L \\ & \forall i : x_{\text{parent}(i)} \geq x_i \\ & \forall i : x_i \in \{0, 1\}, \end{aligned}$$

where $F(e_i)$ is the score of e_i. We define $F(e_i)$ as follows:

$$F(e_i) = \frac{\sum_{w \in W(e_i)} \text{tf}(w, D)}{\text{Depth}(e_i)},$$

where $W(e_i)$ is the set of words contained in e_i. tf(w, D) is the term frequency of word w in a document D. Depth(e_i) is the depth of e_i in the DEP-DT.

3 Tree Knapsack Problem with Dependency-based Discourse Parser

3.1 Motivation

In (Hirao et al., 2013), they automatically obtain the DEP-DT by transforming from the parsed RST-DT. We simply followed their method for obtaining the DEP-DTs [1]. The transformation algorithm can be found in detail in (Hirao et al., 2013). Figure 1(a) shows an example of the RST-DT. According to RST, a document is represented as a tree whose terminal nodes correspond to elementary discourse units (EDUs) and whose non-terminal nodes indicate the role of the contiguous EDUs, namely, 'nucleus (N)' or 'satellite (S)'. Since a nucleus is more important than a satellite in terms of the writer's purpose, a satellite is always a child of a nucleus in the RST-DT. Some discourse relations between a nucleus and a satellite or two nuclei are defined.

Since the TKP of (Hirao et al., 2013) employs a DEP-DT obtained from an automatically parsed RST-DT, their method strongly relies on the accuracy of the RST parser. For example, in Figure 1(a), if the RST-DT parser incorrectly sets the node n_2 as Satellite and the node n_3 as Nucleus, we obtain an incorrect DEP-DT in Figure 1(b) because the transformation algorithm uses the Nucleus-Satellite relationships in the RST-DT. The dependency relationships in Figure 1(b) are quite different from that of the correct DEP-DT in Figure 1(c). In this example, the parser failed to determine the most salient EDU e_2, that is the root EDU of the gold DEP-DT. Thus, the summary extracted from this DEP-DT will have a low ROUGE score.

The results motivated us to design a new discourse parser fully trained on the DEP-DTs and

[1] Li et al. also defined a similar transformation algorithm (Li et al., 2014). In this paper, we follow the transformation algorithm defined in (Hirao et al., 2013).

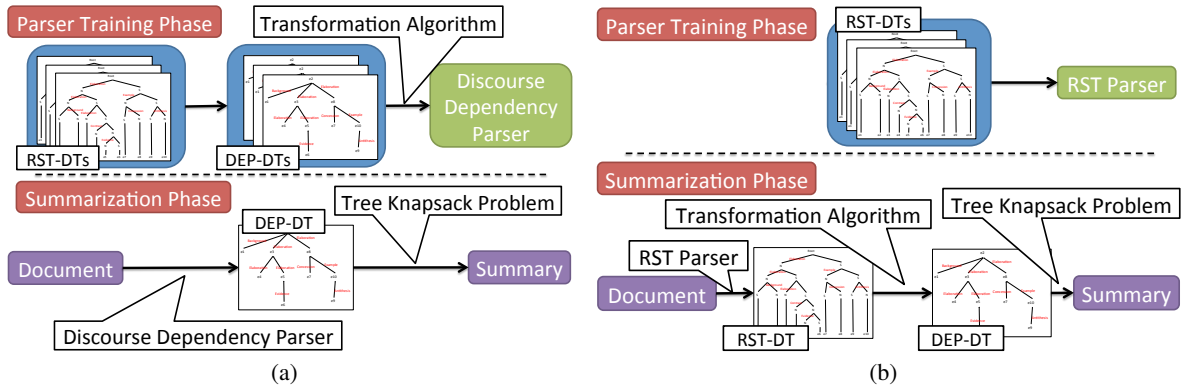


Figure 2: (a) Overview of our proposed method. In the parser training phase, the parser is trained on the DEP-DTs, and in the summarization phase, the document is directly parsed into the DEP-DT. (b) Overview of (Hirao et al., 2013). In the parser training phase, the parser is trained on RST-DTs, and in the summarization phase, the document is parsed into the RST-DT, and then transformed into the DEP-DT.

that could directly generate the DEP-DT. Figure 2(a) shows an overview of the TKP combined with our DEP-DT parser. In the parser training phase, we transform RST-DTs into DEP-DTs, and directly train our parser with the DEP-DTs. In the summarization phase, our method parses a raw document directly into a DEP-DT, and generates a summary with the TKP.

3.2 Description of Discourse Dependency Parser

Our parser is based on the first-order Maximum Spanning Tree (MST) algorithm (McDonald et al., 2005b). Our parser extracts the features from the EDU e_i and the EDU e_j. We use almost the features as those shown in (Hernault et al., 2010). **Lexical N-gram features** use the beginning (or end) lexical N-grams ($N \in \{1, 2, 3\}$) in e_i and e_j. We also include POS tags for the beginning (or end) lexical N-grams ($N \in \{1, 2, 3\}$) in e_i and e_j. **Organizational features** include the distance between e_i and e_j. They also include the number of tokens, and features for identifying whether or not e_i and e_j belong to the same sentence (or paragraph). Soricut et al. (2003) introduced **dominance set features**. They include syntactic labels and the lexical heads of head and attachment nodes along with their dominance relationship. We cannot use the **strong compositionality features** and **rhetorical structure features** described in (Hernault et al., 2010) because we have to know the subtree structures in advance when using these features.

To train the parser, we choose the Margin Infused Relaxed Algorithm (MIRA) (McDonald et al., 2005a; Crammer et al., 2006). We denote $s(\mathbf{w}, \mathbf{y}) = \mathbf{w}^T \mathbf{f}_\mathbf{y}$ as a score function given a weight vector $\mathbf{w}$ and a DEP-DT $\mathbf{y}$. $L(\mathbf{y}, \mathbf{y}^\star)$ is a loss function, and we define it as the number of EDUs that have an incorrect parent EDU in a predicted DEP-DT $\mathbf{y}^\star = \arg\max_{\mathbf{y}} s(\mathbf{w}, \mathbf{y})$. Then, we solve the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w}} \ ||\mathbf{w} - \mathbf{w}^{(t)}|| \\ &\text{s.t.} \ \ s(\mathbf{w}, \mathbf{y}) - s(\mathbf{w}, \mathbf{y}^\star) \geq L(\mathbf{y}, \mathbf{y}^\star), \end{aligned} \tag{1}$$

where $\mathbf{w}^{(t)}$ is a weight vector in the t-th iteration.

3.3 Redesign of Loss Function for Tree Knapsack Problem

When we make a summary by solving a TKP, we do not necessarily need a DEP-DT where all of the parent-child relationships are correct. This is because we rarely select the EDUs around the leaves in the DEP-DT. On the other hand, the parent-child relationships around the root EDU in the DEP-DT are important because we often select the EDUs around the root EDU. Incorporating these intuitions enables us to develop a DEP-DT parser optimized for the TKP. To incorporate this information, we define the following loss function:

$$L_{\text{Depth}}(\mathbf{y}, \mathbf{y}^\star) = \sum_{(i,r,j) \in \mathbf{y}} \frac{[1 - I(\mathbf{y}^\star, i, j)]}{\text{Depth}(e_i)}, \tag{2}$$

where $I(\mathbf{y}^\star, i, j)$ is an indicator function that equals 1 if EDU e_j is the parent of EDU e_i in the

DEP-DT $\mathbf{y}^\star$ and 0 otherwise. In Section 4, we report results with the original loss function $L(\cdot,\cdot)$ and with the modified loss function $L_{\text{Depth}}(\cdot,\cdot)$.

4 Experimental Evaluation

4.1 Corpus

We used the RST-DT corpus (Carlson et al., 2002) for our experimental evaluations. The corpus consists of 385 Wall Street Journal articles with RST annotation, and 30 of these documents also have one human-made reference summary. We used these 30 documents as the test documents for the summarization evaluation, and used the remaining 355 RST annotated documents as the training data for the parser. Note that we did not use the 30 test documents for the summarization evaluation when we trained the parser.

4.2 Summarization Evaluation

We compared the following three systems that differ in the way they obtain the DEP-DT.

TKP-GOLD Used a DEP-DT converted from a gold RST-DT.

TKP-DIS-DEP Used a DEP-DT automatically parsed by our discourse dependency-based parser (DIS-DEP). Figure 2(a) shows an overview of this system.

TKP-DIS-DEP-LOSS Used a DEP-DT automatically parsed by our discourse dependency-based parser (DIS-DEP). Figure 2(a) shows an overview of this system. It is trained with the loss function defined in equation (2).

TKP-HILDA Used a DEP-DT obtained by transforming a RST-DT parsed by HILDA, a state-of-the-art RST-DT parser (Hernault et al., 2010). Figure 2(b) shows an overview of this system.

Hirao et al. (2013) proved that TKP-HILDA outperformed other methods including Marcu's method (Marcu, 1998), a simple knapsack model, a maximum coverage model and LEAD method that simply takes the first L tokens (L = summary length). Thus, we only employed TKP-HILDA as our baseline.

We follow the evaluation conditions described in (Hirao et al., 2013). The number of tokens in each summary is determined by the number in the human-annotated reference summary. The average length of the reference summaries corresponds to about 10% of the words in the source document. This is also the commonly used evaluation condition for single-document summarization evaluation on the RST-DT corpus. We employed the recall of ROUGE-1, 2 as the evaluation measures.

	ROUGE-1	ROUGE-2
TKP-GOLD	0.321	0.112
TKP-DIS-DEP	0.319	0.109
TKP-DIS-DEP-LOSS	0.323	0.121
TKP-HILDA	0.284	0.093

Table 1: ROUGE Recall scores

Table 1 shows ROUGE scores on the RST-DT corpus. We can see TKP-DIS-DEP and TKP-DIS-DEP-LOSS outperformed TKP-HILDA, and achieved almost the same ROUGE scores as TKP-GOLD. Wilcoxon's signed rank test in terms of ROUGE rejected the null hypothesis, "there is a difference between TKP-HILDA and TKP-DIS-DEP (or TKP-DIS-DEP-LOSS)" (Wilcoxon, 1945). This would be because test documents are relatively small.

We analyzed the differences between the proposed systems (TKP-DIS-DEP and TKP-DIS-DEP-LOSS) and TKP-HILDA. First, we evaluated the overlaps between the EDUs in summaries generated by the system and the EDUs in summaries generated by TKP-GOLD. To see the overlaps, we calculated the average F-value using Recall and Precision defined as follows: Recall $= |S_s \cap S_g|/|S_g|$, Precision $= |S_s \cap S_g|/|S_s|$, where S_s is a set of EDUs in a summary generated by a system, and S_g a set of EDUs in a summary generated by TKP-GOLD. The first line in Table 2 shows the results. TKP-DIS-DEP and TKP-DIS-DEP-LOSS outperformed TKP-HILDA as regards the average F-values. The result revealed that TKP-DIS-DEP and TKP-DIS-DEP-LOSS have more EDUs in common with TKP-GOLD than TKP-HILDA. This result is evidence that TKP-DIS-DEP and TKP-DIS-DEP-LOSS outperformed TKP-HILDA in terms of ROUGE score.

Second, we evaluated the root accuracy (RA), the rate at which a parser can find the root of DEP-DTs. Since the root of a gold DEP-DT is the most salient EDU in a document, it should be included in the summary. The second line in Table 2 shows that our methods succeeded in extracting the root

	TKP-DIS-DEP	TKP-DIS-DEP-LOSS	TKP-HILDA
Avg F-value	$0.532^{\star}$	$0.532^{\star}$	0.415
RA	$0.933^{\star}$	$0.933^{\star}$	0.733
Avg DAS	$0.847^{\star}$	$0.843^{\star}$	0.596

$\star$: significantly better than TKP-HILDA ($p < .05$)

Table 2: Average F-value, Root Accuracy (RA), and average Dependency Accuracy in Summary (DAS). Wilcoxon's signed rank test in terms of average F-value, RA and DAS accepted the null hypothesis.

TKP-GOLD:
Elcotel Inc. expects fiscal second-quarter earnings to trail 1988 results. **Elcotel, a telecommunications company, had net income of $272,000, or five cents a share, in its year-earlier second quarter.** The lower results, Mr. Pierce said. Elcotel will also benefit from moving into other areas. Elcotel has also developed an automatic call processor. Automatic call processors will provide that system for virtually any telephone, Mr. Pierce said, not just phones.

TKP-DIS-DEP, TKP-DIS-DEP-LOSS:
Elcotel Inc. expects fiscal second-quarter earnings to trail 1988 results. Elcotel, a telecommunications company, had net income of $272,000, or five cents a share, in its year-earlier second quarter. George Pierce, chairman and chief executive officer, said in an interview. Although Mr. Pierce expects that line of business to strengthen in the next year. Elcotel will also benefit from moving into other areas. Elcotel has also developed an automatic call processor.

TKP-HILDA:
Elcotel Inc. expects fiscal second-quarter earnings to trail 1988 results. **That several new products will lead to a "much stronger" performance in its second half.** George Pierce, chairman and chief executive officer, said in an interview. Mr. Pierce said Elcotel should realize a minimum of $10 of recurring net earnings for each machine each month. Elcotel has also developed an automatic call processor. Automatic call processors will provide that system for virtually any telephone.

Figure 3: Summaries of wsj_2317. The sentences shown in bold-face are the root EDUs in each DEP-DT of the summary.

of DEP-DT with high accuracy.

Third, to evaluate the coherency of the generated summaries, we compared the average Dependency Accuracy in Summary (DAS), which is defined as follows:

$$\mathrm{DAS}(S) = \frac{1}{|S|} \sum_{e \in S} \delta(e),$$

$$\delta(e) = \begin{cases} 1 & (\text{if parent}(e) \in S) \\ 0 & (\text{otherwise}), \end{cases}$$

where S is a set of EDUs contained in the summary and parent(e) returns the parent EDU of e in the gold DEP-DT. DAS(S) measures the rate of the correct parent-child relationships in S. When DAS equals 1, the summary is a rooted subtree of the gold DEP-DT. The third line in Table 2 shows the results. The results demonstrate that the summaries generated by TKP-DIS-DEP or TKP-DIS-DEP-LOSS tend to preserve the upper level dependency relationships between the EDUs within the gold DEP-DT.

Figure 3 shows summaries of wsj_2317 generated by the three systems. The EDUs corresponding to the root of the DEP-DT are used in each system shown in boldface. We can see that the root EDU in the gold DEP-DT is found in the summaries generated by TKP-DIS-DEP and TKP-DIS-DEP-LOSS, but not in the summary generated by TKP-HILDA.

5 Conclusion

In this paper, we proposed a novel dependency-based discourse parser for single-document summarization. The parser enables us to obtain the DEP-DT without transforming the RST-DT. The evaluation results showed that the TKP with our parser outperformed that with the state-of-the-art RST-DT parser, and achieved almost equivalent ROUGE scores to the TKP with the gold DEP-DT.

References

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2002. Rst discourse treebank, ldc2002t07.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

Hal Daumé III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 449 – 456, Philadelphia, PA, July 6 – 12.

Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. 2010. Hilda: a discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3).

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on EMNLP*, pages 1515–1520.

Yuta Kikuchi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2014. Single document summarization based on nested tree structure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 315–320, Baltimore, Maryland, June. Association for Computational Linguistics.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35, Baltimore, Maryland, June. Association for Computational Linguistics.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu. 1998. Improving summarization through rhetorical parsing tuning. In *Proc. of The 6th Workshop on VLC*, pages 206–215.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, December.