# 係り受け木に基づく談話構造の提案

吉田 康久　　鈴木潤　　平尾 努　　永田 昌明
日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
yoshida.y@lab.ntt.co.jp

## 1　はじめに

文書の談話構造を捉えるための理論として修辞構造理論 (Rhetorical Structure Theory; RST)[1] や Discourse Tree Adjoining Grammar (D-TAG)[2] や Cross-document Structure Theory (CST)[3] などがあり，それらに基づきアノテーションしたコーパスとして，RST には RST discourse treebank[4]，D-TAG には Penn Discourse Tree Bank (PDTB) [5]，CST には CST corpus[3] がある．自然言語処理では応用によって利用したい談話構造の情報が異なるため，複数の談話構造理論が存在することは好ましい．実際，RST は機械翻訳 [6]，D-TAG は対話 [7]，CST は要約 [3] に利用されている．

このような背景を踏まえて，我々は係り受け構造木に基づく談話構造の表現方法を提案する．我々の談話構造の表現方法は RST を元にしており，節や文などのテキストユニットをノード，それらの間の修辞関係をエッジに持つ係り受け木として談話構造を表現する．我々はこれを 'Document-level Dependency Representation' (DDR) と呼び，DDR に基づく談話構造木のことを DDR discourse tree (DDR-DT) と呼ぶ．DDR-DT の例を図 1 に示す．図 1 では $e_1$ から $e_{10}$ はほぼ節に相当する Elementary Discourse Unit (EDU) であり，$e_0$ は根を表わすノードである．例えば $e_9$ は $e_{10}$ を 'Antithesis' の関係で修飾しているということを表わしている．

DDR-DT はこのように係り受け木を用いた表現方法であることから，親子関係を利用しテキストユニット間の相対的な顕著性を表現することができる．また，部分木から文書がどのような意味的なまとまりに分割されるかを表現することもできる．前者の例では，$e_5$ は $e_6$ より相対的に重要であることが分かり，後者の例では図 1 の文書は $e_1$，$e_3$-$e_6$，$e_7$-$e_{10}$ の 3 つの意味的なまとまりに分割できることが分かる．

こうした DDR-DT を得るため，本稿では RST に基づく談話構造木 (RST Discourse Tree; RST-DT) から DDR-DT へ一意に変換する方法を提案する．DDR-DT は RST-DT の派生として捉えることができ，RST-DT の利用者は必要に応じてどちらかを使い分けることができる．この変換方法は言語や分野に依存しないため，中国語やスペイン語の RST コーパス [8, 9] やマニュアル文書の RST コーパス [10] など様々な RST コーパスに適用することができ，汎用性が高い．

DDR-DT の持つ特徴を明らかにするため，RST コーパスから変換した DDR-DT のコーパスから係り受けの距離や修辞関係ラベルの統計量と解析器の解析精度の二つの側面を調査した．その結果，DDR-DT は標準的な文内の単語から単語への係り受けよりも長距離の係り受けが多く，また DDR-DT の解析は文に対する単語の係り受け解析よりもはるかに難しいタスクであることが分かった．

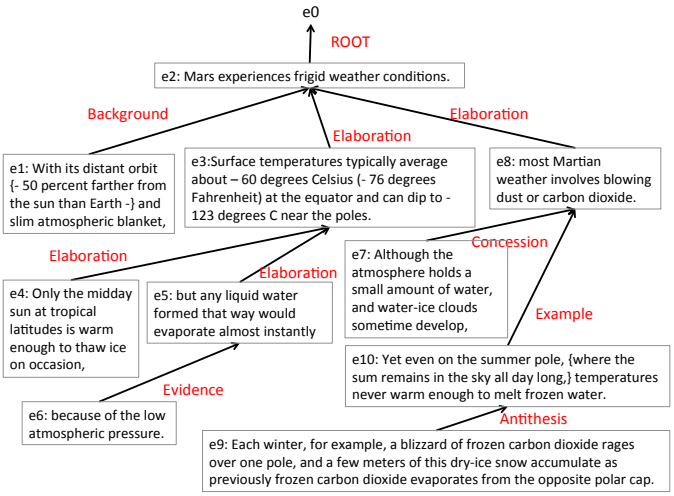


図 1: DDR-DT の例．文献 [11] より引用した RST-DT を変換した．

## 2　関連研究

RST は文書の意味的関係を記述するために作られた理論であり，文書を修辞関係ラベルとともに階層的構造によって表現する．図 2 に RST-DT の例を示す．RST-DT では，EDU と EDU が結合され新たな一つのノードとなり，ノードとノードも全体で一つの木になるまで結合される．EDU やノードが結合される際に修辞構造ラベルが付与される．その際にそれぞれの EDU やノードに Nucleus(核) か Satellite(衛星) のラベルも付与される．Nucleus は Satellite よりも相対的に重要である．PDTB は D-TAG を元に作られたコーパスである [2]．PDTB では 'because' や 'but' などの接続表現に対する項としてテキストユニットを取り，それらの間の談話構造を二項関係で表わす．CST は複数文書間の談話構造を表わす理論であり，文書をまたいだテキストユニット間の関係を表わす．ただし，木構造ではない．

RST-DT と DDR-DT は文書内全体の談話構造を表現するが，PDTB は文書内の文間あるいは句の間の局所的な談話構造のみを表現している．PDTB は D-TAG に基づいており，述語項構造をよく似た構造を持ち，項の間の修辞関係ラベル付きの二項関係を表わしている．一方，RST は終端を EDU として再帰的にテキストスパンが組み上げられた，句構造木に近い consistency の形で表わされている．

このように，各談話構造理論は表現する範囲やその方法が異なっている．しかし，これらの中には (1) 文

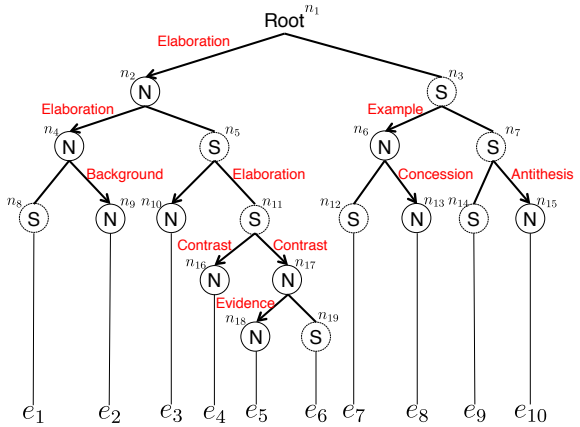


図 2: RST-DT の例．N は Nucleus，S は Satellite を表わす，n1 から n19 までの丸は RST のノードを表わす．

書全体の談話構造を表わし，(2) テキストユニット間を二項関係で表わす，談話構造理論がまだ存在しない．我々の提案する DDR はこれまであった談話構造理論を補完する談話構造の表現方法であり，上記の二点を満たすものである．(1) については DDR では 1 章で述べたようなノード間の親子関係による顕著性や部分木を用いた文書の一貫性を表現することができる．また，(2) については DDR ではテキストユニット間の修辞関係ラベルを直接知ることができるので，例えば評判分析ではレビュアーが商品を評価した「理由」が記述を知ることができるようになる．このようなテキストユニット間の二項関係は，評判分析だけでなく質問応答や対話などの言語処理応用にも有用である．

## 3 RST-DT から DDR-DT への変換方法

本章では RST-DT から DDR-DT への変換方法を説明する．この変換方法は文献 [12] の head finding rule に基づいているが，我々はこれを修辞関係ラベルを含むように精緻化したものである．

まず，$\boldsymbol{t}$ を与えられた文書の RST-DT とし，$\boldsymbol{t}$ は $N$ 個の EDU を持つとする．$\boldsymbol{t}$ の第 $i$ 番目の EDU を $e_i$ と表わす (ただし，$1 \leq i \leq N$)．さらに，$\boldsymbol{R}$ を RST で事前に定義された修辞関係のラベル集合とする．ここで，DDR-DT を定義するため，まず，ラベル付き有向辺を以下のように定義する．

**Definition 1** (ラベル付き有向辺)．$i$ と $j$ を *EDU* のインデックスとし，$r$ を修辞関係のラベルとする．このとき，$i$ 番目の *EDU* から $j$ 番目の *EDU* へのラベル付き有向辺は $(i, r, j)$ の三つ組を用いて定義される．

さらに DDR-DT に変換する際に仮想的な根の EDU $e_0$ を導入し，修辞関係ラベル 'ROOT' を追加する．この仮想的な根 $e_0$ への 'ROOT' ラベルに向かって枝が張られるときのみ，つまり $(i, \mathrm{ROOT}, 0)$ のときのみ出現することに注意されたい (ただし，$1 \leq i \leq N$)．

最後に DDR-DT を以下の条件を満たすラベル付き有向辺の集合として定義する．

1. $(i, r, j) \in \boldsymbol{y}'$, where $1 \leq i \leq N$, $1 \leq j \leq N$, $i \neq j$, and $r \in \boldsymbol{R}$
2. $(i, \mathrm{ROOT}, 0) \in \boldsymbol{y}''$, where $1 \leq i \leq N$
3. $\boldsymbol{y} = \boldsymbol{y}' \cup \boldsymbol{y}''$, where $|\boldsymbol{y}| = N$
4. $\boldsymbol{y}$ has a directed path from for all EDUs $e_i$, where $1 \leq i \leq N$, to the dummy root EDU $e_0$.

**Algorithm 1** convert-rst-into-dep

**Require:** RST-DT $\boldsymbol{t}$
**Ensure:** DDR-DT $\boldsymbol{y}$
$\boldsymbol{y} \leftarrow \emptyset$
**for all** EDU $e_i$ in $\boldsymbol{t}$ **do**
  $i \leftarrow \mathrm{Index}(\boldsymbol{t}, e_i)$
  $P \leftarrow$ find-Node-NearestNucleus($\boldsymbol{t}$, $i$)
  **if** isRoot($P$) = TRUE **then**
    $r \leftarrow$ ROOT
    $j \leftarrow 0$
  **else**
    $r \leftarrow \mathrm{Label}(P)$
    $P \leftarrow \mathrm{Parent}(P)$
    $j \leftarrow$ find-EDU-LMNucleusPath($\boldsymbol{t}$, $P$)
  **end if**
  $\boldsymbol{y} \leftarrow \boldsymbol{y} \cup (i, r, j)$
**end for**
Return $\boldsymbol{y}$

**Algorithm 2** find-Node-NearestNucleus($\boldsymbol{t}$, $i$)

**Require:** RST-DT $\boldsymbol{t}$, index of EDU $i$
**Ensure:** $P$
$P \leftarrow \mathrm{node}(\boldsymbol{t}, i)$
**while** isNucleus($P$) = TRUE and isRoot($P$) = FALSE **do**
  $P \leftarrow \mathrm{Parent}(P)$
**end while**
Return $P$

**Algorithm 3** find-EDU-LMNucleusPath($\boldsymbol{t}$, $P$)

**Require:** RST-DT $\boldsymbol{t}$, node in RST-DT $P$
**Ensure:** $j$
**while** isLeaf($P$) = FALSE **do**
  $P \leftarrow \mathrm{LeftmostNucleusChild}(P)$
**end while**
$j \leftarrow \mathrm{Index}(P)$
Return $j$

Algorithm 1 に RST-DT から DDR-DT への変換手続きを示す．Algorithm 1 中の関数，find-Node-NearestNucleus，find-EDU-LMNucleusPath をそれぞれ Algorithm 2，3 に示す．これらのアルゴリズムにおいて Parent($P$) はノード $P$ の親ノードを返す関数，Label($P$) はノード $P$ への修辞関係ラベルを返す関数，Index($P$) はノード $P$ の直下にある EDU のインデックスを返す関数，LeftmostNucleusChild($P$) はノード $P$ の最左の Nucleus の子供のノードを返す関数，Index($\boldsymbol{t}, e_i$) は EDU $e_i$ のインデックスを返す関数である．

Algorithm 2 では，現在着目している EDU を $e_i$ としたとき，アルゴリズムは RST-DT の根に向かって最も近い Satellite のノードを探す．このとき得られたノード $P$ が RST-DT の根だった場合 (この場合は根まで辿ってきた全てのノードが Nucleus であった，ということである)，$e_i$ は直接擬似的な根の EDU を親とする (修辞関係ラベルは 'ROOT' ラベルを振る)．そうでない場合は，$P \leftarrow \mathrm{Parent}(P)$ とし，再び葉ノードに辿りつくまで $P$ の Nucleus で最左の子ノードをたどっていく．最後に Algorithm 3 では見つけた葉ノードの下にある EDU のインデックスを $e_i$ の親として返す．

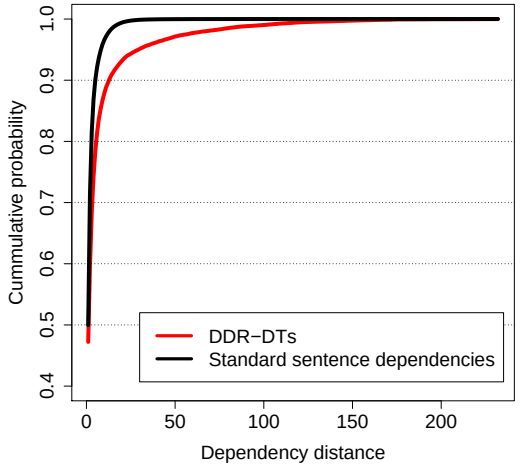


図 3: 係り受けの距離の累積確率分布.

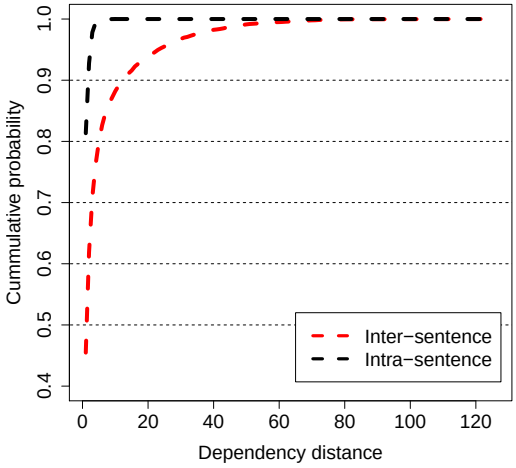


図 4: 文内と文間の係り受けの距離の累積確率分布.

## 4 コーパス統計量による DDR-DT の特徴付け

本章では DDR-DT の特徴を明らかにするために，標準的な文内の単語から単語への係り受け木 (ここでは「文の係り受け木」と呼ぶ) と DDR-DT の違いについてコーパス統計量を通して議論する．

文の係り受け木には Penn Treebank を Penn2Malt で変換したものを利用する．DDR-DT には RST Discourse Corpus[4] 中の RST-DT を第 3 章で説明した変換したものを利用する．このコーパスは Penn TreeBank の一部の 385 記事に対して修辞構造理論のアノテーションがされたものである．修辞関係ラベルとしては大分類の 18 種類を利用する．

文の係り受け木の係り受けの距離の平均，最大値，95%信頼区間はそれぞれ 2.66，218，[1，9] となり，DDR-DT の係り受けの距離の平均，最大値，95%信頼区間はそれぞれ 6.84，232，[1, 30] であった．また，図 3 に係り受けの距離の累積確率分布を示す．

図 3 より，DDR-DT は文の係り受け木よりも長距離の係り受けが多いということが分かる．長距離の係り受けは短距離の係り受けよりも難しいということが知られているが，この統計量から DDR の解析は文の係り受け解析よりも難しいということが想像される．

また，図 4 に文内と文間の係り受けの距離の累積確率分布を示す．図 4 より，文間の係り受けは文内の係り受けよりもかなり長距離であると分かる．さらに文内と文間の修辞関係ラベルの分布も調査した．結果を表 1 に示す．表 1 から，文内と文間で修辞関係ラベルの分布がかなり異なることが分かる．例えば，‘Attribution’ は文内に偏って出現しており，‘Topic-Change’ は文間に偏って出現している．

| | Intra-sentence | Inter-sentence |
|---|---|---|
| Attribution | 98.9% (3050) | 1.1% (34) |
| Background | 55.3% (563) | 44.7% (455) |
| Cause | 57.7% (369) | 42.3% (271) |
| Comparison | 65.9% (122) | 34.1% (63) |
| Condition | 85.9% (269) | 14.1% (44) |
| Contrast | 53.9% (372) | 46.1% (318) |
| Elaboration | 49.3% (4426) | 50.7% (4543) |
| Enablement | 95.0% (547) | 5.0% (29) |
| Evaluation | 13.4% (64) | 86.6% (415) |
| Explanation | 24.0% (287) | 76.0% (909) |
| Joint | 0.0% (0) | 100.0% (5) |
| Manner-Means | 89.9% (232) | 10.1% (26) |
| Summary | 33.1% (111) | 66.9% (224) |
| Temporal | 88.8% (207) | 11.2% (26) |
| TextualOrganization | 5.3% (7) | 94.7% (126) |
| Topic-Change | 4.1% (13) | 95.9% (304) |
| Topic-Comment | 19.2% (51) | 80.8% (214) |

表 1: 文内と文間の修辞構造ラベルの分布.

## 5 解析器の解析精度による DDR-DT の特徴付け

次に DDR 解析器の性能に基づき，DDR の特徴を調べる．解析器としては，HILDA，One-step parser，Two-step parser を用いた．HIDLA は文献 [13] で提案された最高精度の RST 解析器である．HILDA から DDR-DT を得るために，まず，HILDA を利用し，生文書から RST-DT を得る．次に，第 3 章で提案した変換方法で DDR-DT へ変換した．One-step parser は Maximum Spanning Trees (MSTs) algorithm を利用した解析器である．One-step parser の素性には HILDA で使用しているの中で One-step parser で使用可能なもののみ使用した[1]．Two-step parser は One-step parser を元にした二段階解析器である．Two-step parser では，文内の解析を行なった後に文間の解析を行なう．まず，文内の解析では EDU の列を入力として受け取り次に EDU から EDU への係り受け木を出力する．次に文間の解析では文の列 (ただし，文には文内における EDU から EDU へ係り受け木が付与されている) を入力として受け取り，文から文への係り受け木を出力する．最後に文から文への係り受け関係を EDU から EDU への係り受け関係に戻す．文内，文間の解析はともに MST algorithm を用いる．Two-step parser では文内と文間の解析器を構築するが，先に述べたようにそれぞれ特徴が異なるので素性も分割して設計する．文内の解析器の素性は One-step parser の素性とほぼ同じものを使うので説明を省略する．文間の解析器の素性としては，Subtree feature[2]や Sentence

[1]例えば，部分木の修辞関係ラベルなどの素性は One-step parser では使用することができない

[2]文間の解析器では解析が終わった文内の部分木の情報を利用．例としては，文内の根の EDU に係る EDU の数や修辞構造ラベル

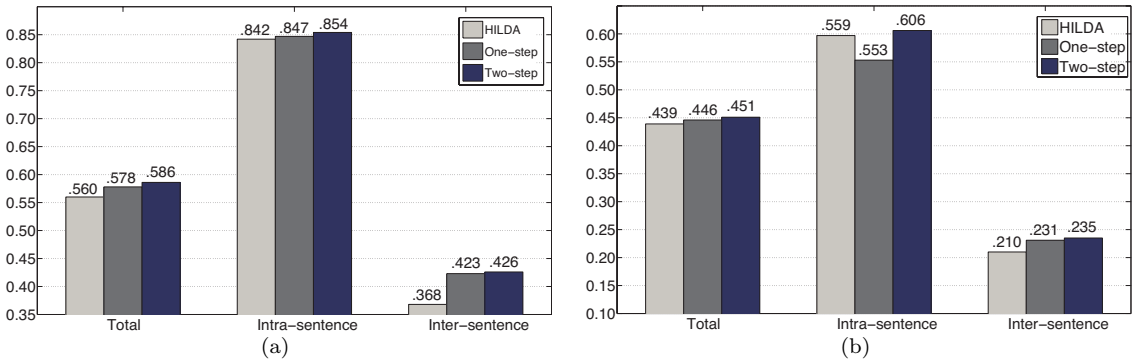


図 5: 各解析器の解析精度．(a) 各解析器の UAS．(b) 各解析器の LAS．

importance feature[3]，Text segmentation feature[4]などを利用した．

実験には，第 4 章と同様に RST Discourse Corpus[4]を利用した．評価指標には，文書中のテキストユニットのうち正しく係り先を見つけることができたものの割合 (Unlabeled Attachment Score; UAS) と文書中のテキストユニットのうち正しく係り先を見つけることができ，かつ修辞関係ラベルも正しいものの割合 (Labeled Attachment Score; LAS) を用いた．さらに，詳細な解析精度を知るために Total Accuracy(文書中に含まれる EDU の UAS と LAS)，Intra-sentence Accuracy(文中に含まれる EDU の UAS と LAS)，Inter-sentence Accuracy(文書中に含まれる文の UAS と LAS) を指標として使用した．

### 5.1 解析精度による DDR の特徴付け

図 5 に各解析器の解析精度を示す．全体の UAS はどの解析器も 6 割を下回っている．文の係り受け解析が 9 割を越えていることを考慮すると，DDR-DT への解析は文の係り受け解析と比較すると非常に難しいタスクであることが分かる．特に文間は文内と比較すると非常に精度が下がっている．要因としては，(1) 第 4 章で述べたように DDR-DT の係り受けは文の係り受けと比較すると長距離であること (文間の関係は長距離のものが特に多い)，(2) 文の係り受け解析における品詞情報に相当するものが DDR-DT には現状存在しないこと，の二点が挙げられる．

## 6 まとめ

本稿では RST に基づいた談話構造の係り受け木による表現，DDR を提案した．DDR は，(1) 文書全体の談話構造を表わしているため，談話の顕著性や一貫性を表現でき，(2) テキストユニット間の二項関係で表わせるため，ユニット間の修辞関係を直接表現することができる，という利点があり，これは既存の談話構造理論を補完するものである．また，DDR-DT 持つ特徴をコーパス統計量と解析器の解析精度の二つの側面から調査した．その結果，DDR-DT は文の係り受け木よりも長距離の依存関係を持っておりそのため解析が非常に難しいこと，また文内と文間で係り受け距離や修辞関係ラベルの分布が異なり，特に文間の解析が難しいことが分かった．今後は DDR の解析器の精度向上を目指すとともに，DDR を用いることで自然言語処理応用の性能向上も目指していきたい．

を利用する．

[3]左右の文の TF-IDF やページランク．

[4]TextTiling のセグメント間の距離やスコア．

## 参考文献


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

[2] Katherine Forbes, Eleni Miltsakaki, Rashmi Prasad, Anoop Sarkar, A. Joshi, B. Webber, Aravind Joshi, and Bonnie Webber. D-ltag system: Discourse parsing with a lexicalized tree adjoining grammar. *Journal of Logic, Language and Information*, 12:261–279, 2002.

[3] Dragomir Radev, Jahna Otterbacher, and Zhu Zhang. Cstbank: Cross-document structure theory bank. http://tangra.si.umich.edu/clair/CSTBank, 2003.

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

[5] Rashmi Prasad, Aravind Joshi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, and Bonnie Webber. The penn discourse treebank as a resource for natural language generation. In *In Proc. of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*, pages 25–32, 2005.

[6] Mei Tu, Yu Zhou, and Chengqing Zong. A novel translation framework based on rhetorical structure theory. In *Proc. of the 51st ACL*, pages 370–374, 2013.

[7] Sara Tonelli, Giuseppe Riccardi, Rashmi Prasad, and Aravind K. Joshi. Annotation of discourse relations for conversational spoken dialogs. In *LREC*, 2010.

[8] Ming Yue. Rhetorical structure annotation of chinese news commentaries. *Journal of Chinese Information Processing*, 22(04):19–24, 2008.

[9] Iria Cunha, Juan-Manuel Torres-Moreno, and Gerardo Sierra. On the development of the rst spanish treebank. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 1–10, 2011.

[10] Rajen Subba and Barbara Di Eugenio. An effective discourse parser that uses rich linguistic information. In *Proc. of HLT: The 2009 NAACL*, pages 566–574, 2009.

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

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

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