修 士 論 文

決定木学習を応用した日本語要約文の自動作成

指導教官　奥村学 助教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

原口良胤

1999 年 2 月 15 日

要 旨

近年，電子化されたテキストが氾濫し，必要な情報を得るために多大な労力が要求されるようになってきている．そこで，必要な情報のみを効率良く抽出することが可能な，自動要約技術の必要性がますます高まってきている．

決定木学習を要約作成に利用するには，要約をある文書中の重要文の集合ととらえ，人手によってあらかじめ重要文を決定する．この文集合から，各文についてその特徴を属性値として計算し，それらの属性値を用いて訓練を行い決定木を構成する．次に要約の対象となる文書の各文について属性値を計算し訓練によって得られた決定木を用いて重要文を抽出し要約を作成する．

決定木学習は，あらかじめ用意された複数のクラスに分類された訓練事例をもとに決定木を構成し，それによって新たな事例を複数のクラスに分類する．ここで，あるクラスの事例数が他のクラスの事例数と比較して極めて少ないと，それらの事例はノイズとして扱われ，うまく分類できない．要約作成を行う場合，重要文は非重要文と比べて事例数が極端に少ないので，決定木学習をそのまま用いると重要文の事例がうまく分類されない．

本研究では，2 段階の決定木を構成する手法 (TLDT:Two Level Decision Tree construction) を適用することで，決定木学習を要約作成に適用する際の問題を解消することを試みる．TLDT では，訓練事例中の重要文の属性値と類似の属性値を持つ非重要文 (ニアミス) を抽出し，重要文と合わせて新たなクラス mix を構成する．一方，重要文の属性値と類似していない属性値を持つ非重要文でも新たなクラス far を構成する．1 段階目の決定木学習では，これら 2 つのクラスをよく分離する決定木を構成する．2 段階目の決定木学習では，クラス mix を重要文クラスと非重要文クラスに分離するような決定木を構成する．

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1 研究の背景と目的

決定木学習を要約作成に利用しようという試みはいくつか行われている．[野本 97] らは，日経新聞の記事をもとに，人手で重要文の決定を行ったコーパスを訓練データとして C4.5 アルゴリズムによる決定木学習を行い，重要文の抽出を行っている．また [Aone97] らが開発した要約システムでも C4.5 決定木学習アルゴリズムを用いている．しかし，これらの試みでは決定木学習を行う際，重要文抽出のような片方のクラスに含まれる事例数がもう片方のクラスに含まれる事例数に比べて極端に少ない場合の C4.5 アルゴリズムの適用方法に関して特別の考慮はされていない．本研究では，このような場合の C4.5 アルゴリズム適用に [本田 96] らの手法を利用することによって，精度の向上を目指す．

C4.5 決定木学習アルゴリズムでは，クラス間の事例数に大きな偏りがあり，あるクラスに属する事例数が統計的に誤差と考えられてしまうほど少ないような場合，信頼性のあるモデルが構築できないことがある．本田らは，こういった場合にも信頼性のあるモデルが得られるような手法を提案し，テキストセグメンテーションに適用し，良い結果を得た．

2 つのクラスがあるとき，事例数の多い方のクラスに含まれる，事例数の少ない方のクラスの類似事例をニアミスとして抽出し，事例数の少ない方のクラスと結合して新たなクラスを作る．これと，事例数の多い方のクラスの残りの事例とを合わせて訓練事例とし，学習を行い決定木を構成する．次に，事例の少ない方のクラスとニアミスを分類するような決定木を構成する．この 2 段階の決定木によって，事例数の少ないクラスを抽出することができる．

この結果，C4.5 をそのまま適用する場合と比較して，精度の向上が見られた．この手法は，テキストセグメンテーションだけでなく，重要文抽出を行う場合にも適用すること

ができると考えられる．

1.1.1 C4.5 決定木学習アルゴリズム

C4.5 決定木学習アルゴリズム [Quinlan93] は，分類クラスといくつかの属性をもつ訓練事例の集合から決定木を構成するシステムである．決定木は以下の構成要素からなる．

- 葉:クラスを示す.
- 判別ノード:1 つの属性がどのような属性値をとるかを調べるテストを指定する．テストのそれぞれの結果に 1 つの分岐と部分木が対応する．

決定木は，根ノードから葉にはじめて出会うまでテストを繰り返しながら事例を分類する．中間ノードでは事例のテストが行われ，このテストの結果にしたがって部分木の根に進む．このプロセスが最終的に葉まで到達した時，その事例のクラスは葉に記録されているクラスであると判定される．訓練事例は，あらかじめ決められた属性とその属性値によって記述され，前もって定義された分類クラスに割り当てられている．決定木の生成には分割統治法を用い，根ノードから後戻りなしに決定木を生成する．基本的なアルゴリズムを以下に示す.

Step 1 すべての事例を根ノードに配置する.

Step 2 そのノードが持つ事例集合が単一のクラスに属すならば，**Step 4** へ，そうでなければ **Step 3** へ.

Step 3 以下の手順で子を作成する.
自分が持つ事例集合をもっとも良く区別する属性を 1 つ選択する．この属性が現在のノードにおけるテストの対象となる．選択した属性の属性値に従って事例を部分集合に分割する．それぞれの部分集合にノードを付与し，親ノードとの間に属性値の名前を持つリンクを張る．作成したノードすべてに対して **Step 2** を実行する.

Step 4 それぞれのノードにおいて，部分木に分割した時の予測される誤り数と，そのノードをもっとも大きな事例の部分集合を持つ部分木に置き換えた時の予測される誤り数を計算し，部分木が存在する時の予測される誤り数の値が大きければ，そのノードを枝刈りして，もっとも大きな事例を持つ部分木に置き換える.

1.1.2 決定木学習を用いた要約作成

決定木学習を用いて要約作成を行う場合，あらかじめ人手で作成した重要文集合を訓練事例とする．このときテキストを何らかの方法で訓練可能な事例にする必要がある．そこで，[Paice90]，[Edmundson69] らが示したように，テキストに含まれる文ごとに属性を設定し，さらにそれらを複数組み合わせることによって文の重要度を評価する．これらの文ごとに計算された属性データと，人手で選択された重要文を訓練事例として決定木学習を行う．

1.1.3 本研究の目的

本研究では，決定木学習を自動要約作成に適用する際に問題となる，事例数の偏りを，非重要文事例からニアミスを抽出し重要文事例と結合，新たなクラスに設定しなおすことによって解消する．更に重要文とニアミスをよく分離することのできる 2 段目の決定木を構成することによって，C4.5 決定木学習を用いた要約作成の精度の向上を計る．

1.2 本論文の構成

本論文では，2 章で決定木学習を用いた要約手法を紹介し，本研究との比較を行う．3 章では決定木学習を行う際に用いるテキストの属性について説明を行い，4 章では TLDT の説明を行う．5 章では行った実験について述べる．6 章では 5 章で行った実験について考察を行い，7 章，8 章ではそれぞれ，まとめと今後の課題について述べる．

第 2 章

決定木学習を用いた要約

C4.5 決定木学習を用いた要約手法を図 2.1に示す.

2.1 決定木学習を用いた要約の関連研究

2.1.1 野本の手法

[野本 97] では，C4.5 決定木学習を用い，複数の属性を組み合わせて，あらかじめ重要文を選択されたテキストから重要文の抽出を行っている．対象としたテキストは 1995 年の日経新聞の記事，社説，コラムである．ここで利用されている属性は，表 2.1の通りである.

表 2.1: 使用する属性

分野	文の属する分野 (随想，社説，報道)
テキスト内位置	文章内の文の位置
見出しとの類似度	文と文章の見出しとの類似度
テキスト内 $tf \cdot idf$	文の内容的独立度
態度表現の型	態度表現の型
文の長さ	文の文字数
段落内位置	段落における文の位置

図 2.1: C4.5 による要約

2.1.2 Aone の手法

[Aone97] らは C4.5 を用いて，人手で作成された重要文集合を訓練事例として決定木学習を行い，要約のための重要文抽出を行っている．

2.1.3 Mani の手法

[Mani98] らは，決定木学習を利用した要約作成を行う際の，重要文の事例数と非重要文の事例数に極端な偏りがあることによる重要文抽出精度の低下を回避する試みを行っている．テキスト中で重要文を決定，抽出した後，それとほぼ同数の非重要文をテキストからランダムに抽出し，それらを合わせて訓練事例として決定木学習を行っている．

2.2 関連研究と本研究との比較

[野本 97] ら，[Aone97] らは，C4.5 による決定木学習を行う際に，重要文の事例数と非重要文の事例数の大きな偏りによる重要文抽出精度の低下の可能性に対して，とくに考慮をしていない．一方，[Mani98] では，事例数の偏りを解消して精度の向上を試みているが，非重要文から重要文とほぼ同数の事例を抽出する際には，特別な知識は使用せず，ランダムに行っている．

本研究はこれらの手法と異なり，非重要文から事例を抽出する際には重要文と属性値が類似しているものとしてニアミスという概念を用いる．

第 3 章

テキストの属性

3.1 テキストの属性

本研究で用いるテキストの属性は表 3.1の通りである.

表 3.1: 使用する属性

情報分類	抽出方法	属性値の種類
文の位置	先行する文数/文書内の総文数	連続値
文の長さ	文の文字数/文書内の最長文の文字数	連続値
段落内位置	段落内先行文数/段落内総文数	連続値
$tf \cdot idf$	文の $tf \cdot idf$	連続値
態度表現の型	文末の型	13 値
語彙的連鎖	文中に含まれる語彙的連鎖	連続値
助詞	主格の助詞の種類	3 値
接続詞 1	後続する文の文頭の接続詞の種類	11 値
接続詞 2	先行する文の文頭の接続詞の種類	11 値
前方照応	文頭に出現する照応詞の種類	4 値

それぞれの属性について以下に述べる.

3.1.1 文の位置

テキストは, ジャンルに依存する構造上の規則性がある程度存在すると考えられる. このため, 文の出現する位置によって重要度をある程度予測することが可能である [Edmundson69].

本研究では対象となる文のテキスト内での位置を表す．テキスト内で i 番目の文 s_i の属性値 loc_i は，テキスト内の総文数 $|S|$ とすると，

$$loc_i = \frac{i-1}{|S|} \tag{3.1}$$

3.1.2 文の長さ

[Kupiec95] 対象とする文の文字数を表す．対象とする文を s_i とすると，属性値 loc_i は，

$$len_i = |s_i| \tag{3.2}$$

3.1.3 段落内位置

対象とする文が含まれている段落内における，先行する文数を表す．ある段落 p_j において i 番目の文 s_i の属性値 $ploc_i$ は，

$$ploc_i = \frac{i-1}{|p_j|} \tag{3.3}$$

3.1.4 $tf \cdot idf$

文の内容的独立度を表す．対象とする文 s_i に含まれている語 w の $tf \cdot idf$ 値を $tf \cdot idf(w)$ とすると，属性値 $tf \cdot idf(i)$ は，

$$\begin{aligned} tf \cdot idf(i) &= \sum_{w \in W(s_i)} tf \cdot idf(w) \\ tf \cdot idf(w) &= tf(w) \cdot idf(w) \end{aligned}$$

ここで，$W(s_i)$ は，文 s_i に含まれる語の集合を示す．また，$tf(w)$ は対象テキスト内での語の w 頻度，$idf(w)$ は，$df(w)$ を w が出現したテキストの総数とすると，

$$idf(w) = \frac{\log \frac{N}{df(w)}}{\log N} \tag{3.4}$$

3.1.5 態度表現の型

文末の表現を元に，

叙述	判断	断定	推量	要望	願望	伝達	義務	理由	問いかけ	命令	体言止め	その他

の 13 値のいずれかを属性値として決定する．

3.1.6 語彙的連鎖

語彙的連鎖 [Morris91] とは，テキスト中で意味的つながり (語彙的結束性) を持つ語の連なりのことである．1 つの連鎖はテキスト中のある話題を表すと考えられるため，重要な連鎖に属す単語を多く含む文がより重要であると考えることで文の重要度が決定できる．テキストに含まれるある語とその類義語が構成する語彙的連鎖をテキストから抽出し，語の出現頻度に基づいてスコアリングを行う．連鎖 c の重要度 $weight_c$は，$|c|$ を連鎖 c を構成する単語数とすると，

$$weight_c = |c| \tag{3.5}$$

と表される．次に，対象とする文に含まれている語彙的連鎖の重要度を合計し，テキスト内の重要度の最大値で正規化したスコアを属性値とする．

$$score_{s_i} = \frac{\sum_{c \in s_i} weight_c}{MAX(weight_c)} \tag{3.6}$$

3.1.7 助詞

対象となる文に含まれる主格の助詞の種類を示す．属性値は，

「は」	「が」	なし

の 3 値である．

3.1.8 接続詞

対象となる文に後続する文の文頭に出現する接続詞の種類を示す．属性値は，

逆説	順接	説明補足	添加	話題転換	選択	常識提示	後ろ強調	並列	曖昧	なし

3.1.9 前方照応

対象となる文に出現する前方照応の種類を示す．属性値は，

「あ」型	「こ」型	「そ」型

の 3 値である．

第 4 章

2 段階決定木構成手法 TLDT

4.1 TLDT

TLDT の概要を図 4.1に示す.

C4.5 決定木学習アルゴリズムを用いて決定木学習を行うとき，事例数に偏りのある訓練集合を用いて小数事例を抽出する決定木を構築しようとしても，予測力の低い木しか獲得できない．本手法は，ウインドウ処理 [Quinlan93] の初期ウインドウの選択に似た手法をとる．ウインドウ処理では，それぞれのクラスに属す事例数が同数になるように初期ウインドウを選択し，決定木を構築する．本手法も事例数がほぼ同数になるように事例を選択し決定木を構成するが，2 つの決定木を構成する．

TLDT のアルゴリズムは次のようなステップで行われる．ここで，訓練事例の集合を S，S中のすべての事例は 2 つのクラスに属し，事例数の少ない方のクラスを c_s，多い方のクラスを c_lとする．c_lに属す事例の集合を C_l，c_sに属す事例の集合を C_sとする．

$$C_l \cup C_s = S \tag{4.1}$$

$$C_l \cap C_s = \emptyset \tag{4.2}$$

それぞれの事例は，離散属性，連続属性のいずれも取り得る．取り得る属性の集合を $A = \{A_1, A2, ..., A_n\}$ とする．

1. 属性の集合 A の部分集合 $A'(A' \subset A)$ を取った時，C_sに含まれる事例と，A'の属性値がすべて同じ属性値を持つ C_lの部分集合 $C'_l(C'_l \subset C_l)$ を $|C'_l| \approx |C_s|$ となるように取り出す．$|C'_l|$ 中の事例をクラス c_sのニアミスと呼ぶ.

TLDT(Two Level Decision Tree construction)の概要

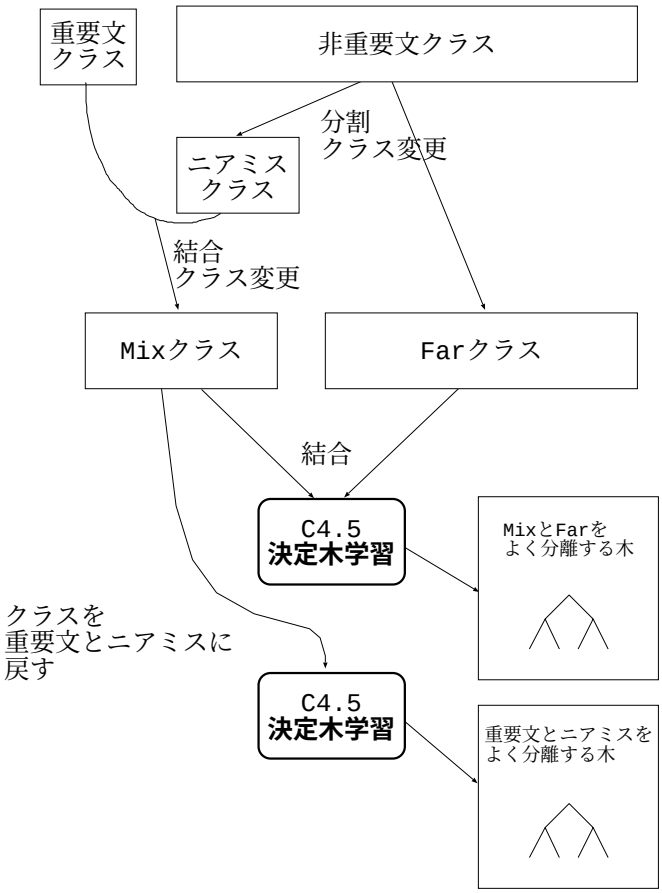

図 4.1: TLDT の概要

2. S中の事例のクラスの変更を行う．c_sに属す事例とニアミスクラスの事例のクラスをc_{mix}に，c_lに属すがニアミスでない事例のクラスをc_{far}に変更する．c_{mix}に属す事例の集合をC_{mix}，c_{far}に属す事例の集合をC_{far}とする．

$$C_{mix} = \{x | x \in C_s \cup C'_l\} \tag{4.3}$$

$$C_{far} = \{x | x \in C_l \cap \overline{C'_l}\} \tag{4.4}$$

3. C_{mix}と，C_{far}を訓練事例として，最初の決定木$dt1$を構築する．

4. C'_l中の事例のクラスをc'_lに変更し，C_s，C'_lを訓練事例として，2 つ目の決定木$dt2$を構築する．

4.2 ニアミス決定方法

ニアミスの決定は4.1で述べたように行われるが，必ずしもすべての属性値が同じ$|C_l'| \approx |C_s|$となるようなC_l'中の事例，つまりニアミスを取れるとは限らない．そこで本研究では，以下の2通りの方法を用いる．

1. C_lに属する文で，C_sに属する文と同じ属性値を持つ属性の個数が多い文から順にスコアを付与し，そのスコアの上位から順に，$|C_l'| \approx |C_s|$となるようにニアミスC_l'の事例を取り出す．またこのとき，連続値を持つ属性に関しては離散値に近似することによって，より多くのニアミスを選択することができる．

2. C_lに属する文で，C_sに属するある1つの文と同じ属性値を持つ属性のもっとも多い文を1つ抽出する．C_sに属するすべての文について1つのニアミスを，重複のないように選択する．

第 5 章

実験

5.1 対象文書

実験では，1995 年の日経新聞のコラム，社説，記事の 3 種類をそれぞれ 25 本，計 75 テキスト，総文数 1424 文を使用した．これは，計 112 人の大学 (院) 生を被験者とし，重要だと思う文をテキスト中の文の総数の 10%程度テキストから選択させてある．

表に，コーパスの概要を示す．表中の平均文数と平均重要文数は，それぞれ 1 テキスト当たりのものである．

表 5.1: 対象コーパスについて

総文数	総重要文数	平均文数	平均重要文数
1424	202	18.99	2.69

5.2 重要文抽出実験 1

本実験は，

- ニアミス判定時に使用する属性
  全属性，一部の属性，離散属性のみ
- 決定木 $dt1$ 訓練時の mix クラスと far クラスの比
  1:1，1:2
- 決定木を構成する時の属性値
  離散属性のみ，連続属性を含む

の 3 通りの条件を組み合わせ，$n = 10$ の交差検定を行った．

5.2.1 評価方法

実験の評価には，*recall*，*precision*，*f-measure* を用いた．*recall*，*precision*，*f-measure* はそれぞれ，

$$
\begin{aligned}
recall &= \frac{\text{決定木が抽出した文に含まれる重要文数}}{\text{重要文数}} \\
precision &= \frac{\text{決定木が抽出した文に含まれる重要文数}}{\text{決定木が抽出した文数}} \\
f\text{-}measure &= \frac{2 \cdot precision \cdot recall}{recall + precision}
\end{aligned}
$$

である．

5.2.2 重要文抽出実験 1 結果

表 5.2では，5.2で述べた組み合わせによって得られた TLDT で構築される決定木 dt1，dt2 を組み合わせたシステムの出力，およびプロダクションルール pr1，pr2 の組み合わせたシステムの出力を，C4.5 をそのまま使用した結果，[Mani98] の手法による結果と比較して示す．

表 5.2: 離散属性による決定木 (mix:far=1:1)

ニアミス判定		recall	precision	f-measure
なし	C4.5 dt	0	0	0
	C4.5 pr	0.233	0.395	0.293
全属性 (TLDT)	dt2	0.609	0.221	0.324
	pr2	0.645	0.210	0.317
	dt1-dt2	0.385	0.267	0.315
	pr1-pr2	**0.433**	**0.264**	**0.328**
一部属性 (TLDT)	dt2	0.610	0.219	0.322
	pr2	0.617	0.198	0.300
	dt1-dt2	0.391	0.295	0.336
	pr1-pr2	**0.389**	**0.274**	**0.322**
なし	Mani dt	0.583	0.232	0.332
	Mani pr	**0.637**	**0.225**	**0.333**

表 5.3: 離散属性による決定木 (mix:far=1:2)

ニアミス判定		recall	precision	f-measure
なし	C4.5 dt	0	0	0
	C4.5 pr	0.233	0.395	0.293
全属性 (TLDT)	dt2	0.609	0.221	0.324
	pr2	0.645	0.210	0.317
	dt1-dt2	0.218	0.418	0.287
	pr1-pr2	**0.335**	**0.345**	**0.340**
一部属性 (TLDT)	dt2	0.610	0.219	0.322
	pr2	0.617	0.198	0.300
	dt1-dt2	0.219	0.294	0.251
	pr1-pr2	**0.272**	**0.290**	**0.281**
なし	Mani dt	0.365	0.433	0.396
	Mani pr	**0.451**	**0.361**	**0.401**

表 5.4: 連続属性による決定木 (mix:far=1:1)

ニアミス判定		recall	precision	f-measure
なし	C4.5 dt	0.218	0.565	0.315
	C4.5 pr	0.222	0.461	0.300
全属性 (TLDT)	dt2	0.627	0.223	0.329
	pr2	0.616	0.204	0.306
	dt1-dt2	0.395	0.302	0.342
	pr1-pr2	**0.362**	**0.282**	**0.317**
一部属性 (TLDT)	dt2	0.656	0.204	0.311
	pr2	0.600	0.192	0.291
	dt1-dt2	0.437	0.308	0.361
	pr1-pr2	**0.395**	**0.279**	**0.327**
離散属性 (TLDT)	dt2	0.622	0.207	0.311
	pr2	0.573	0.205	0.302
	dt1-dt2	0.461	0.277	0.346
	pr1-pr2	**0.459**	**0.297**	**0.361**
なし	Mani dt	0.659	0.233	0.344
	Mani pr	**0.538**	**0.294**	**0.380**

表 5.5: 連続属性による決定木 (mix:far=1:2)

ニアミス判定		recall	precision	f-measure
なし	C4.5 dt	0.218	0.565	0.315
	C4.5 pr	0.222	0.461	0.300
全属性	dt2	0.627	0.223	0.329
	pr2	0.616	0.204	0.306
	dt1-dt2	0.367	0.418	0.391
	pr1-pr2	**0.280**	**0.454**	**0.347**
一部属性	dt2	0.656	0.204	0.311
	pr2	0.600	0.192	0.291
	dt1-dt2	0.312	0.340	0.326
	pr1-pr2	**0.259**	**0.405**	**0.316**
離散属性	dt2	0.622	0.207	0.311
	pr2	0.573	0.205	0.302
	dt1-dt2	0.354	0.378	0.365
	pr1-pr2	**0.279**	**0.499**	**0.358**
なし	Mani dt	0.403	0.316	0.354
	Mani pr	**0.383**	**0.356**	**0.369**

5.3 重要文抽出実験 2

5.2で行なった実験の結果を考慮して，更に以下のような実験を行なった．

1. C4.5 の使用属性の拡張
2. Mani の手法の条件を変更
3. TLDT の条件を変更
4. プロダクションルールの確信度を用いる
5. 要約率を考慮した TLDT の拡張

これらそれぞれの実験について $n = 10$ の交差検定を行った．

5.3.1 実験手法

C4.5 の使用属性の拡張

表 3.1で示した 9 属性を用いる．ある属性値列を，計算の対象となった文の前後それぞれ 1 文に対しても同様に付与し，合計 27 属性を用いて C4.5 決定木学習を行う．

Mani の手法の条件を変更

Mani の手法において，非重要文集合からランダムにサンプリングする非重要文数は，重要文とほぼ同じである．この実験では，重要文とランダムに抽出する非重要文の比をそれぞれ 1:1，1:2 に設定し，C4.5 決定木学習を行う．

TLDT の条件を変更

TLDT において，ニアミスの比較方法，ニアミスの抽出方法，mix クラスの事例数と far クラスの事例数を以下の表 5.6のとおり用意し，それぞれを組み合わせて合計 8 通りの条件で実験を行う．

表 5.6: TLDT の実験条件

ニアミス比較方法	全属性使用	離散属性使用
ニアミス抽出方法	集合で抽出	事例ごとに抽出
mix:far	1:1	1:2

プロダクションルールの確信度を用いる

決定木学習によって得られたプロダクションルールの確信度の高いものから要約率に応じて正解を抽出する．プロダクションルールの確信度とは，そのルールを用いて分類を行う際に正解となる確率である．本実験では，TLDT，MANI の手法それぞれにおいて，確信度の高いものから上位 10%を抽出し，正解とする．

要約率を考慮した TLDT の拡張

要約率に応じた正解数を得るために，多クラスあるいは多段階の決定木を構築する．本実験では以下の 2 通りの手法を採用した．

1. 要約率 10%以下，10%から 20%，20%から 30%，非重要文，の 4 クラスで 1 段階の決定木を構築する．それぞれの要約率に対応した正解を分類する木を構築し，複数

のクラスの正解を結合することによって，30%以下の要約率について，10%きざみで正解数を得る．

2. 重要文，非重要文の 2 クラスで要約率 30%以下の決定木，要約率 20%以下の決定木，要約率 10%以下の 3 段階の決定木を構築する．3 段階に構築された決定木によって分類された結果を，要約率に応じて結合することによって，30%以下の要約率について，10%きざみで正解数を得る．

5.3.2 評価方法

実験の評価には，5.2.1と同様，*recall*，*precision*，*f-measure* を用いた．

5.3.3 重要文抽出実験 2 結果

以下に 5.3に示した条件で行なった実験の結果を示す．

表 5.7: C4.5 のみ:属性ごと

属性	dt/pr	要約率	recall	precision	f-measure
文の位置	dt	8.8/142.4=0.062	5.1/21.2=0.241	5.1/ 8.8=0.580	0.340
	pr	9.0/142.4=0.063	5.3/21.2=0.250	5.3/ 9.0=0.589	0.351
文の長さ	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	1.7/142.4=0.012	0.6/21.2=0.028	0.6/ 1.7=0.353	0.052
段落内位置	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
tfidf	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	2.3/142.4=0.016	1.0/21.2=0.047	1.0/ 2.3=0.435	0.085
態度表現の型	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
語彙的連鎖	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	0.8/142.4=0.006	0.2/21.2=0.009	0.2/ 0.8=0.250	0.018
助詞	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
接続詞	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
前方照応	dt	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
	pr	0.0/142.4=0.000	0.0/21.2=0.000	0.0/ 0.0=0.000	0.000
全 9 属性	dt	8.3/142.4=0.058	4.7/21.2=0.222	4.7/ 8.3=0.566	0.319
	pr	9.4/142.4=0.066	4.9/21.2=0.231	4.9/ 9.4=0.521	0.320
全 27 属性	dt	10.4/142.4=0.073	4.3/21.2=0.203	4.3/10.4=0.413	0.272
	pr	8.1/142.4=0.057	4.7/21.2=0.222	4.7/ 8.1=0.580	0.321

表 5.8: C4.5 Mani 手法:全属性:重要文/非重要文比による変化

重要文/非重要文比	dt/pr	要約率	recall	precision	f-measure
1:1	dt	58.2/142.4=0.409	13.0/21.2=0.613	13.0/58.2=0.223	0.327
	pr	34.5/142.4=0.242	10.6/21.2=0.500	10.6/34.5=0.307	0.381
1:2	dt	28.1/142.4=0.197	9.2/21.2=0.434	9.2/28.1=0.327	0.373
	pr	20.9/142.4=0.147	8.6/21.2=0.406	8.6/20.9=0.411	0.409

表 5.9: TLDT:複数の条件を組み合わせた実験

実験条件		要約率	recall	precision	f-measure
全属性, 集合,1:1	dt1-dt2	0.221	0.415	0.291	0.342
	pr1-pr2	0.226	0.436	0.296	0.353
	dt2	0.414	0.589	0.214	0.314
	pr2	0.437	0.624	0.215	0.320
全属性, 集合,1:2	dt1-dt2	0.055	0.220	0.611	0.324
	pr1-pr2	0.065	0.206	0.504	0.292
	dt2	0.221	0.427	0.303	0.355
	pr2	0.224	0.396	0.266	0.318
全属性, 事例,1:1	dt1-dt2	0.199	0.368	0.283	0.320
	pr1-pr2	0.109	0.280	0.414	0.334
	dt2	0.413	0.523	0.191	0.280
	pr2	0.199	0.366	0.301	0.330
全属性, 事例,1:2	dt1-dt2	0.093	0.245	0.395	0.302
	pr1-pr2	0.073	0.243	0.503	0.327
	dt2	0.141	0.294	0.320	0.307
	pr2	0.116	0.289	0.395	0.334
離散属性, 集合,1:1	dt1-dt2	0.246	0.473	0.287	0.358
	pr1-pr2	0.176	0.375	0.332	0.352
	dt2	0.418	0.590	0.212	0.312
	pr2	0.374	0.577	0.242	0.341
離散属性, 集合,1:2	dt1-dt2	0.056	0.234	0.645	0.343
	pr1-pr2	0.060	0.232	0.527	0.322
	dt2	0.228	0.443	0.295	0.354
	pr2	0.182	0.375	0.315	0.342
離散属性, 事例,1:1	dt1-dt2	0.206	0.414	0.302	0.349
	pr1-pr2	0.151	0.363	0.378	0.370
	dt2	0.383	0.601	0.230	0.333
	pr2	0.360	0.552	0.258	0.351
離散属性, 事例,1:2	dt1-dt2	0.099	0.262	0.416	0.321
	pr1-pr2	0.057	0.192	0.507	0.279
	dt2	0.168	0.376	0.341	0.357
	pr2	0.137	0.328	0.358	0.343

表 5.10: 要約率が 10%になるように確信度が高いものから抽出

実験条件		要約率	recall	precision	f-measure
離散属性, 事例,1:1	pr1-pr2	0.091	0.107	0.129	0.116
	pr2	0.362	0.561	0.231	0.327
MANI	pr	0.097	0.539	0.379	0.439

表 5.11: 要約率に対応した拡張

実験条件		要約率	recall	precision	f-measure
4 クラス 1 段	pr	0.017	0.091	0.148	0.113
	dt	0.018	0.119	0.240	0.159
2 クラス 3 段	pr1-pr2-pr3	0.012	0.100	0.100	0.100
	dt1-dt2-dt3	0.015	0.000	0.000	0.000

第 6 章

考察

6.1 重要文抽出実験 1

TLDT 手法による重要文抽出実験について，ここでは 5.4の，決定木学習の訓練事例での mix クラスと far クラスの比率が 1:1 で連続属性による決定木構築の結果について考察する．TLDT 手法による重要文抽出の結果，C4.5 単独と比較して，すべての場合で recall が向上している．一方，precision はすべての場合において低下している．ここで f-measure を見ると，すべての場合において向上が見られており，全体としての精度はわずかだが向上していることがわかる．また，mix クラスと far クラスの比を大きくすると，recall の低下，precision の向上，全体として f-measure が若干向上する傾向が見られる．

次に，決定木学習によって得られた決定木を考察する．

1 段目の決定木の例 (一部)

```
文の位置 <= 0 : M (62.0/1.4)
文の位置 > 0 :
|   文の位置 <= 0.91 :
|   |   接続詞 = opp: F (42.0/4.9)
|   |   接続詞 = ord: F (8.0/2.4)
|   |   接続詞 = swt: F (0.0)
|   |   接続詞 = sel: F (1.0/0.8)
|   |   接続詞 = css: M (3.0/2.1)
|   |   接続詞 = emp: F (4.0/1.2)
|   |   接続詞 = pal: F (1.0/0.8)
|   |   接続詞 = exp:
|   |   |   tfidf <= 0.28 : M (2.0/1.0)
|   |   |   tfidf > 0.28 : F (8.0/2.4)
|   |   接続詞 = add:
|   |   |   前方照応 = ko: F (2.0/1.0)
|   |   |   前方照応 = so: M (2.0/1.0)
|   |   |   前方照応 = aa: F (0.0)
|   |   |   前方照応 = nn: F (28.0/3.7)
|   |   接続詞 = amb:
|   |   |   tfidf <= 0.82 : F (14.0/2.5)
```

1 段目の決定木では，上に示したように，まず属性「文の位置」が 0 以下かどうかによって大きく mix クラスと far クラスが分離されている．次に「文の位置」が 0.91 以下かどうかが検査され，そうであれば属性「接続詞」によって多くの場合は far クラスに分類されている．交差検定の他の場合の決定木においても，同様に，まず「文の位置」によって大きく分類され,「接続詞」によって多くは far クラスに分類される．

2 段目の決定木の例 (一部)

```
tfidf <= 0.1 : N (32.0/4.9)
tfidf > 0.1 :
|   前方照応 = ko: Y (13.0/2.5)
|   前方照応 = so: Y (5.0/1.2)
|   前方照応 = aa: Y (0.0)
|   前方照応 = nn:
|   |   tfidf > 0.96 : Y (27.0/4.9)
|   |   tfidf <= 0.96 :
|   |   |   文の長さ <= 0.55 :
|   |   |   |   接続詞2 = ord: Y (2.0/1.0)
|   |   |   |   接続詞2 = exp: N (0.0)
|   |   |   |   接続詞2 = add: Y (2.0/1.0)
|   |   |   |   接続詞2 = swt: N (0.0)
|   |   |   |   接続詞2 = sel: N (0.0)
|   |   |   |   接続詞2 = css: N (0.0)
|   |   |   |   接続詞2 = emp: N (0.0)
|   |   |   |   接続詞2 = pal: Y (1.0/0.8)
|   |   |   |   接続詞2 = amb: Y (2.0/1.0)
|   |   |   |   接続詞2 = opp:
|   |   |   |   |   tfidf <= 0.17 : N (2.0/1.0)
```

2 段目の決定木では，まず属性「$tf \cdot idf$」が 0.1 以下かどうかによって分類され，そうでなければ「前方照応」によって分類される．交差検定の他の場合でも 2 段目の木では，多くの場合まず属性「$tf \cdot idf$」によって大きく分類されている．

6.2 重要文抽出実験 2

6.2.1 C4.5 の使用属性の拡張

対象とする文の前後それぞれ 1 文に，対象とする文と同じ属性値列を付与し 27 属性を用いて実験を行った結果，9 属性のみで実験を行った場合とを比較すると，決定木による分類では f-measure が 15%以上低下した．一方，プロダクションルールによる分類では有意な差は見られなかった．

Mani の手法の条件を変更

Mani の手法において，重要文数と非重要文集合からランダムにサンプリングする非重要文数の比をそれぞれ 1:1，1:2 に設定し，C4.5 決定木学習を行った．その結果，サンプリング数を 2 倍にすることによって recall は低下するものの，precision は向上し，f-measure も 10%前後向上した．

TLDT の条件を変更

TLDT において，ニアミスの比較方法，ニアミスの抽出方法，mix クラスの事例数と far クラスの事例数を組み合わせて合計 8 通りの条件で実験を行ったが，TLDT と比較して有意な差の見られる結果は得られなかった．

プロダクションルールの確信度を用いる

決定木学習によって得られたプロダクションルールの確信度の高いものから要約率に応じて正解を抽出し，要約率に応じた正解数を得ることができた．f-measure は，TLDT においては著しく低下したが，Mani の手法においては向上した．

要約率を考慮した TLDT の拡張

要約率 10%以下，10%から 20%，20%から 30%，非重要文，の 4 クラスで 1 段階の決定木を構築すし，10%の以下のクラスについて評価した．要約率は 2%程度で f-measure は著しく低下した．

重要文，非重要文の 2 クラスで要約率 30%以下の決定木，要約率 20%以下の決定木，要約率 10%以下の 3 段階の決定木を構築し，10%以下のクラスについて評価した．要約率は 1%強で，f-measure は，プロダクションルールが 10%，決定木では 0 となった．

第 7 章

まとめ

要約作成に C4.5 決定木学習を適用する際の問題を指摘し，それを軽減するための TLDT 手法を示した．また，実際に TLDT を実装し，複数の条件を組み合わせて実験を行った．その結果，C4.5 決定木学習を単独で利用するよりも若干精度が向上することを示した．

上記の結果を踏まえ，実験 2 を行った．その結果，TLDT では条件を変化させても重要文を抽出する精度の有意なは向上は見られなかった．一方，Mani の手法では，非重要文のサンプリング数を変化させることで f-measure が向上した．非重要文集合から非重要文をサンプリングすることで正解数が多くなり，要約率が上昇してしまうが，プロダクションルールの確信度を利用することで，与えられた要約率に従った正解を得ることができた．

第 8 章

今後の課題

決定木学習によって得られるプロダクションルールの確信度を用いて，要約率 10%の場合，10%の正解を抽出することができた．この結果に基づき，任意の要約率に応じた正解抽出を行うシステムの構築が考えられる．

謝辞

本研究を進めるにあたり，終始熱心な御指導を賜わりました奥村学助教授に心から感謝致します.

中間審査などの折には，諸先生方から貴重な御意見を頂きましたことを感謝致します.

さらに，貴重な御意見，討論をしていただいた島津明教授，自然言語処理学講座 大石亨助手，ならびに研究室の皆様に感謝致します.

最後に，多くの方々の御援助によって本研究を行うことができましたことを厚く御礼申し上げます.

参考文献

[Aone97] C. Aone , M.E. Okurowski, J. Gorlinsky, and B. Larsen. “A scalable summarization system using robust nlp”. ACL’97. pp. 66–73. 1997.

[Edmundson69] Edmundson, H. “New methods in automatic abstracting”. Journal of ACM, 16 (2), 264-285. 1969.

[Kupiec95] Julian Kupiec , Jan Pedersen , Francine Chen. “A Trainable Document Summarizer”. SIGIR’95. pp. 68–73. 1995.

[Mani98] Inderjeet Mani , Eric Bloedorn. “Machine Learning of Generic and User-Focused Summarization”. Proc. of the 15th National Conference on Artificial Inteligence, pp. 821–826. 1998.

[Morris91] Morris, J. Hirst, G. “Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text” Computational Linguistics, 17 (1), pp. 21–48. 1991.

[Paice90] Paice, C. “Constructing Literature Abstracts by Computer: Techniques and Prospects” Information Processing and Management, 26 (1), pp. 171–186. 1990.

[Quinlan93] J.Ross Quinlan “C4.5: Programs for Machine Learning. Morgan Kaufmann,1992”. (邦訳:AI によるデータ解析, 古川康一 監訳, トッパン,1995).

[Watanabe96] Hideo Watanabe. “A Method for Abstracting Newspaper Articles by Using Surface Clues”. COLING’96. pp974–979.1996

[野本 97] 野本忠司, 松本裕治 “人間の重要文判定に基づいた自動要約の試み”. 情処研報 NL120-11. 1997.

[本田 96] 本田岳夫, 望月源, ホー ツー バオ, 奥村学“小数事例抽出を目的とした決定木学習”. 1996.

[望月 98] 望月源，岩山真, 奥村学“抄録を利用した検索”. 言語処理学会第 4 回年次大会併設ワークショップ「テキスト要約の現状と将来」, pp.22-29. 1998.

[本田 93] 本田岳夫, 奥村学“語義曖昧性を考慮した有意な語彙連鎖の生成”. 情報処理学会研究会資料 NL97-14, pp.95-102. 1993.

APPENDIX A