Vol.2013-NL-211 No.4
Vol.2013-SLP-96 No.4
2013/5/23

多段階戦略に基づくテキスト間の意味関係認識：RITE2タスクへの適用

服部 昇平[1,a)]　佐藤 理史[1]　駒谷 和範[1]

概要：本論文では，RITE2 の BC サブタスクと MC サブタスクを解くシステムについて報告する．BC サブタスクは，与えられた 2 つのテキスト間において，含意が成立するかどうかを判定するタスクである．我々のシステムは，このサブタスクを 2 ステップで解く．最初のステップでは，表層類似度に基づいて，デフォールトの判定を行う．次のステップでは，ヒューリスティックを適用することにより，デフォールトの判定を覆す必要があるかどうかを判定する．MC サブタスクは，片方向含意，双方向含意，矛盾，無関係の 4 クラスを判定するタスクである．我々は，BC サブタスク用のシステムに，矛盾検出モジュールを追加することにより，このサブタスクを解くシステムを作成した．システムの性能を評価するフォーマルランにおいて，我々のシステムの性能は，BC サブタスクでは MacroF1 で 78.61(42 システム中 7 位)，MC サブタスクでは 59.95(21 システム中 1 位) であった．

キーワード：RITE2，多段階戦略，表層類似度，文字オーバーラップ率，ヒューリスティック

A Multi-Step Strategy to Recognize Semantic Relations between Sentences in RITE2 Task

Shohei Hattori[1,a)]　Satoshi Sato[1]　Kazunori Komatani[1]

Abstract: This paper describes a system that executes the Binary-Class (BC) subtask and the Multi-Class (MC) subtask in RITE2. The BC subtask is, given a pair of sentences t1 and t2, to recognize whether t1 entails t2. Our system executes this subtask by two steps. The first step assigns a default class by applying a simple rule base on a character-overlap measure. The second step examines necessity of overwriting the default class by applying heuristic rules. The MC subtask is a four-way labeling task to detect (forward / bi-directional) entailment or no entailment (contradiction / independence) in a sentence pair. We have implemented a subsystem by combining the BC-subsystem with an additional module for detecting contradiction. The system was evaluated in the formal run. In the BC subtask, our system achieved 78.61 in MacroF1, which was the seventh place among 42 runs. In the MC subtask, our system achieved 59.95 in MacroF1, which was the first place among 21 runs.

Keywords: RITE2, multi-step strategy, surface similarity, character overlap ratio, heuristic rules

1. はじめに

命題間に定義される含意・等価・矛盾などの関係は，命題的内容を表すテキスト間の意味関係として，拡張的に解釈することが可能である．テキスト間に成立するこのような意味関係を自動認識する技術は，テキストの意味的解釈には不可欠であり，テキストマイニング等の応用において，強く求められている技術である．

この分野は，英語に対しては，RTE (Recognizing Textual Entailment) という評価型ワークショップを中心に，研究が活発に行なわれてきている．日本語や中国語を対象とした評価型ワークショップは，NTCIR9 の RITE [3] として

[1] 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University
[a)] syohei_h@nuee.nagoya-u.ac.jp

初めて実施された後，この成功を受けて，NTCIR10[*1]ではRITE2 [4] が開催された．

一般に，テキスト間の意味関係認識には，複雑な意味処理や大規模なデータベースが必要と言われている [5]．確かに，意味関係を人間と同じように認識するためには，たとえば，2 つのテキストが同じ事態を表すかどうかを判定することが不可欠であり，そのためには，述語項解析や言い換え認識などの意味処理技術が必要である．しかしながら，それらの技術の成熟度は低く，短期間の開発で，高い精度を望むことはできない．

我々は，RITE2 の参加にあたり，現在の技術レベルに即した方針として，「多段階戦略」と呼ぶ方針を採用し，システム開発を行った．この方法では，まず，テキスト間の表層類似度に基づいて，デフォールトの意味的関係を判定する．次に，ヒューリスティックを適用して，デフォールト判定を覆す必要があるかどうかを判定する．このような戦略を用いることで，1 段目でかなりの判定精度を担保しつつ，2 段目のヒューリスティックの発見を容易にすることを目論んだ．

本稿では，まず，2 節で RITE2 について概説する．次に，3 節でシステムの設計方針について述べる．4 節では，含意認識に有効に働く，テキスト間の表層類似度を検討する．5 節と 6 節では，作成したシステムについて述べる．7 節では，RITE2 の開発用データおよび評価用データに対するシステムの性能について述べ，その結果を考察する．

2. RITE2 の概要

RITE2[4] は，テキスト間の含意，換言，矛盾等の検出の技術開発を目的とした評価型ワークショップである．評価型ワークショップでは，通常，まず，正解が付与された開発用データが配布され，参加者は，そのデータを用いてシステムを開発する．この後，システム評価（フォーマルラン）のための，正解が隠されたデータが配布される．参加者は，システムを実行してそのデータに対する答を求め，ワークショップ主催者に提出する．最後に，提出した結果に基づくシステムの性能が，主催者によって公表される．

RITE2 には，日本語テキストを対象とした 5 つのサブタスク，中国語 (簡体字および繁体字) テキストを対象とした 2 つのサブタスクが含まれている．以下では，本稿で扱う，日本語テキストを対象とした 2 つのサブタスクについて説明する．

Binary Class サブタスク (以下，BC サブタスクと略記) は，与えられたテキストペア $\langle t_1, t_2 \rangle$ に対して，t_1 が t_2 を含意するかどうか判定する 2 値分類問題である．分類ラベルは，以下の 2 種類である．

Y (含意あり): t_1 が t_2 を含意する

N (含意なし): t_1 は t_2 を含意しない

これらは，いずれも方向性を持った関係である．

一方，Multi Class サブタスク (以下，MC サブタスクと略記) は，含意の方向，矛盾の検出も含めた，次の 4 種類のラベルを出力する 4 値分類問題である．

F (片方向含意): t_1 が t_2 を含意し，かつ，t_1 が t_2 を含意しない

B (双方向含意): t_1 が t_2 を含意し，かつ，t_1 が t_2 を含意する

C (矛盾): t_1 が t_2 同時に起こり得ない

I (無関係): 上記以外

これらのうち，F のみが方向性を持った関係であり，他は双方向の関係である．

図 1 に，配布された開発データの例を示す．開発データのそれぞれは，テキストペア $\langle t_1, t_2 \rangle$ と，問題番号，および，正解ラベルから構成されている．フォーマルランのデータもこれと同じ形式であるが，正解ラベルは含まれない．

```
<pair id="1" label="Y">
 <t1>プロメーテウスは人類に火を渡し、張り付けにされた。</t1>
 <t2>プロメテウスは人類に火を齎して罰を受けた。</t2>
</pair>
```

図 1 開発用データ例

		表層類似度 高い	表層類似度 低い
意味的類似度	高い	Case1	Case2
意味的類似度	低い	Case3	Case4

図 2 テキストの意味的類似度と表層類似度

3. 設計方針：多段階戦略

一般に，2 つのテキスト間の意味的類似度と表層類似度には，正の相関があると考えられる．含意認識にテキスト間の表層類似度が有効に働く [1][3][2] という事実は，このことの一つの傍証である．

ここで，意味的類似度と表層類似度を，それぞれ，高・低に分割すると，図 2 に示すような 2×2 の表を描くことができる．2 つの類似度には正の相関があるので，多くのテキストペアは，Case1 または Case4 に分類されることになる．

残りの 2 つの箱のうち，Case2 には，意味的類似度が高いが表層類似度は低いテキストペアが分類される．たとえば，テキストペアが言い換えを含み，かつ，その言い換えの表層上の類似度が低い場合，そのようなテキストペアは，Case2 に分類されることになる．

逆に，Case3 には，表層類似度が高いが，意味的類似度が低いテキストペアが分類される．2 つのテキスト間に見られる表層上の小さな差異が，意味的に大きな差異をもたらす場合，そのようなテキストペアは，Case3 に分類されることになる．

表層類似度の計算法を定めれば，Case1/Case3 と

[*1] http://research.nii.ac.jp/ntcir/ntcir-10/

Case2/Case4 を分離することができる．テキストペアの表層類似度の計算法は，これまでに各種の方法が提案されている．この中から最適な方法を選び，2 つのクラスに分割するための最適なしきい値を決めればよい．

これに加え，本研究では，Case3 を Case1 から分離することを目指す．具体的には，テキストペアの表層上の差異に着目し，その差異が意味的に大きな差異をもたらすかどうかを判定する．

4. 含意認識とテキスト間の表層類似度

前節の方針に従い，本節では，どのような表層類似度が，含意関係の認識に有効かを検討する．

4.1 オーバーラップ率

ここでは，表層類似度を定義するための基礎となる，オーバーラップ率を計算する一般関数を定義する．

まず最初に，ある集合 E が与えられたとき，2 つのテキスト t_1 と t_2 において，集合 E の要素がいくつ共通して現れるかを求める．これを，形式的に，次式によって定義する．

$$overlap(\boldsymbol{E};t_1,t_2)=\sum_{x\in\boldsymbol{E}}\min(freq(x,t_1),freq(x,t_2)) \tag{1}$$

ここで，$freq(x,t)$ は，集合 E の要素 x が，t 中に出現する回数を表す．なお，集合 E には，文字の集合，文字 bigram の集合，単語の集合，単語 bigram の集合などを想定する．

この式を用いて，以下に示す 2 つのオーバーラップ率を定義する．

$$overlap\_ratio_B(\boldsymbol{E};t_1,t_2)=\frac{2\,overlap(\boldsymbol{E};t_1,t_2)}{\sum_{x\in\boldsymbol{E}}freq(x,t_1)+\sum_{x\in\boldsymbol{E}}freq(x,t_2)} \tag{2}$$

$$overlap\_ratio(\boldsymbol{E};t_1,t_2)=\frac{overlap(\boldsymbol{E};t_1,t_2)}{\sum_{x\in\boldsymbol{E}}freq(x,t_2)} \tag{3}$$

前者は，t_1 と t_2 の両方の長さで正規化した値であるのに対し，後者は，t_2 の長さのみで正規化した値である．

含意は，方向性をもった関係である．そのため，t_1 から t_2 への含意関係の認識には，後者の式 (3) を用いる．

この式の集合$\boldsymbol{E}$を定めると，特定の形式 (文字，単語) のオーバーラップ率を定義する式となる．例えば，集合$\boldsymbol{E}$として文字集合 ($\boldsymbol{C^1}$) を採用することにより，文字オーバーラップ率 $cor(t_1,t_2)$ を以下のように定義することができる．

$$cor(t_1,t_2)=overlap\_ratio(\boldsymbol{C^1};t_1,t_2) \tag{4}$$

4.2 9 種類の表層類似度の比較

集合$\boldsymbol{E}$として以下の 9 種類の集合を採用し，それぞれの場合の含意認識の精度を比較した．

$\boldsymbol{C^1}$, $\boldsymbol{C^2}$, $\boldsymbol{C^3}$, $\boldsymbol{W^1}$, $\boldsymbol{W^2}$, $\boldsymbol{W^3}$, $\boldsymbol{L^1}$, $\boldsymbol{L^2}$, $\boldsymbol{L^3}$

ここで，$\boldsymbol{C^n}$は文字 n-gram の集合を表し，$\boldsymbol{W^n}$は形態素出現形 n-gram の集合，$\boldsymbol{L^n}$は形態素基本形 n-gram の集合を表す．なお，テキスト t を形態素の列に変換するために，MeCab[*2]+ UniDic[*3]を用いた．

精度比較の調査には，RITE2 の開発用データを用いた．使用したデータを表 1 に示す．全部で 1242 個 (含意関係が成立するもの 613 個，成立しないもの 629 個) のテキストペアを使用した．なお，表 1 の，順方向 (f) は，$\langle t_1,t_2\rangle$ のペアをそのまま使用したことを表し，逆方向 (b) は，t_1 と t_2 を入れ替えて使用したことを意味する．

調査の手順は，次のとおりである．まず，それぞれのテキストペアに対して，9 種類のオーバーラップ率を計算する．次に，それぞれのオーバーラップ率に対して，最適なしきい値を決定する．具体的には，しきい値を 0.01 づつ変化させていき，しきい値以上を **Y**(含意あり)，しきい値未満を **N**(含意なし) と判定したときの正解率が最も高くなる値を求める．

調査結果を表 2 に示す．この表に示すように，集合$\boldsymbol{E}$として文字 1-gram の集合を用いた場合，最も高い正解率 78.3%が得られ，このときのしきい値は 0.69 であった．この結果に基づき，本研究では，表層類似度の計算に，式 (4) で定義される文字オーバーラップ率 $cor(t_1,t_2)$ を使用する．

$cor(t_1,t_2)$ の 0.01 刻みのヒストグラムを図 3 に示す．この図では，上部は **Y** のテキストペアの個数を，下部に **N** のテキストペアの個数を示している．この図から明らかなように，しきい値 0.69 によって，2 つのクラスがきれいに分割されるわけではない．

表 1 調査に使用したデータ

タスク	BC		MC					
ラベル	Y	N	B		F	C	I	
個数	240	371	83		207	65	193	
方向	f	f	f	b	f	f	f	total
含意あり	✓		✓	✓	✓			613
含意なし		✓				✓	✓	629

f : 順方向 , b : 逆方向

表 2 9 種類の表層類似度の含意認識性能

$\boldsymbol{E}$	しきい値	正解数		正解率
		Y	N	
$\boldsymbol{C^1}$	**0.69**	**499/613**	**473/629**	**78.3**
$\boldsymbol{L^1}$	0.66	468/613	471/629	75.6
$\boldsymbol{W^1}$	0.64	452/613	477/629	74.8
$\boldsymbol{C^2}$	0.45	484/613	473/629	77.1
$\boldsymbol{L^2}$	0.35	413/613	487/629	72.5
$\boldsymbol{W^2}$	0.29	443/613	447/629	71.7
$\boldsymbol{C^3}$	0.36	410/613	490/629	72.5
$\boldsymbol{L^3}$	0.19	365/613	472/629	67.4
$\boldsymbol{W^3}$	0.18	355/613	480/629	67.2

[*2] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
[*3] http://www.tokuteicorpus.jp/dist/

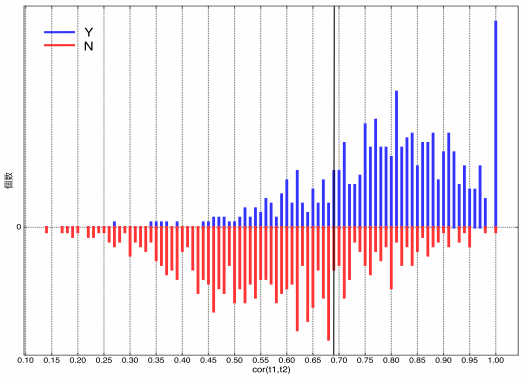

図 3 $cor(t_1, t_2)$ のヒストグラム

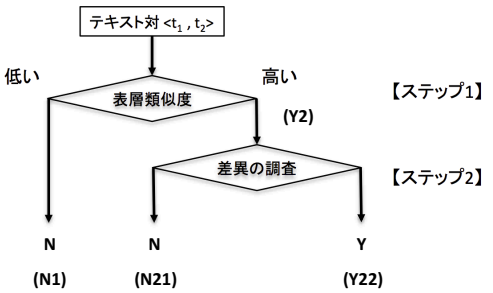

図 4 含意認識システム概要

5. BC サブシステム

5.1 システムの概要

作成した BC サブシステム (含意認識システム) の概要を図 4 に示す．このシステムは，以下の 2 段階のステップで含意認識を行う．

ステップ 1: テキストペアの表層類似度を計算し，類似度が高い場合は **Y**(含意あり)，低い場合は **N**(含意なし) と判定する．

ステップ 2: ステップ 1 で **Y** と判定した場合，そのテキストペアの差異を調査し，その結果を覆す必要があるかどうかを判定する．

以下では，これらのステップについて説明する．

5.2 ステップ 1: 表層類似度に基づく判定

表層類似度に基づく判定は，以下の条件によって行う．

$((cor(t_1, t_2) \geq 0.73)$ or
$(kor(t_1, t_2) > cor(t_1, t_2) \geq 0.69)$ or
$((0.69 > cor(t_1, t_2) > 0.65)$ and
$(kor(t_1, t_2) - 0.1 > cor(t_1, t_2)))$

前節の調査から導かれる条件は，$cor(t_1, t_2) \geq 0.69$ である．しかしながら，しきい値付近では，**Y** と **N** が混在しており，これらをよりうまく分離するために，上記のように修正した．

まず，$cor(t_1, t_2) \geq 0.73$ の場合は，無条件で **Y** と判定する．次に，$0.73 > cor(t_1, t_2) \geq 0.69$ の場合は，追加の条件として，$kor(t_1, t_2) > cor(t_1, t_2)$ を設定する．最後に，$0.69 > cor(t_1, t_2) > 0.65$ の場合は，条件 $kor(t_1, t_2) - 0.1 > cor(t_1, t_2)$ を満たす場合に，**Y** とする．

ここで用いた，$kor(t_1, t_2)$ は，漢字およびカタカナの文字集合 $\boldsymbol{K^1}$ におけるオーバーラップ率を表す．

$$kor(t_1, t_2) = overlap(\boldsymbol{K^1}; t1, t2) \qquad (5)$$

平仮名と比較した場合，漢字とカタカナは，内容語の一部として使用される割合が高い．そのため，$kor(t_1, t_2)$ は，$cor(t_1, t_2)$ よりも，より内容語を重視したオーバーラップ率となる．

なお，上記の条件に含まれるしきい値は，開発用データを用いて実験的に決定した．

5.3 ステップ 2: 差異の調査

ステップ 1 で **Y** と判定した場合，そのテキストペアの差異を調査し，その結果を覆す必要があるかどうかを判定する．本システムでは，固有名詞と数字に着目する．

5.3.1 固有名詞の不整合

含意関係にある 2 つのテキストでは，出現する固有名詞が共通している場合が多い [6]．t_1 から t_2 への含意の判定においては，t_2 に出現するが t_1 には出現しない固有名詞が存在する場合，含意関係が成立しない可能性が高いと考えられる．

このようなヒューリスティックを用いれば，以下に示すテキストペアを，正しく **N** と判定できる．

> t_1: 比較宗教学の観点では世界宗教をエルサレムを聖地とするアブラハムの宗教，インド宗教，東アジア宗教の三つに分類し，他の分野も比較する．
> t_2: 比較宗教学では**イスラム**教，ユダヤ教，キリスト教をアブラハムの宗教に分類する．
> <dataset type = bc(dev.) , id = 44 , label =N>

このペアの文字オーバーラップ率 ($cor(t_1, t_2)$) は 0.76 であるが，t_2 に出現する「イスラム」という固有名詞は，t_1 には出現しない．

上記のヒューリスティックを実現するためには，固有名詞の抽出と照合が必要となる．固有名詞の抽出は，Juman[*4] を利用して t_2 を形態素解析した後，以下のいずれかの条件を満たす形態素を固有名詞として抽出する．

- 「自動獲得:Wikipedia」という素性が付与される形態素
- 「固有名詞」という品詞細分類が付与される名詞

固有名詞の照合，すなわち，抽出した固有名詞 w が t_1 に出現するかどうかの判定は，次の条件で行う．

(a) w と同じ文字列が t_1 に存在する
(b) w から長音符号を削除した文字列が，t_1 から長音符号を削除した文字列中に存在する
(c) w がカタカナ語のとき，t_2 中で w に隣接するカタカナ文字列 w' が t_1 に存在する

ここで，(b) の条件は，カタカナ語の表記ゆれ (長音符号

[*4] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN

の有無) に対処するために導入した．(c) の条件は，主に人名を対処するために導入した (ファーストネームとラストネームのいずれかが存在すればよい)．

5.3.2 数字の不整合

数字も，固有名詞と同様の傾向を示す．開発用データを分析した結果，t_2 には出現するが t_1 には出現しない数字が存在する場合，含意関係が成立しない可能性が高いことが確認された．

数字に対するヒューリスティックを導入すれば，以下に示すテキストペアを，正しく **N** と判定できる．

> t_1: 天京事変は，天京で 1856 年に発生した太平天国の指導部の内紛で，東王楊秀清・北王韋昌輝・燕王秦日綱が命を落とし，2 万人余りが殺害された．
> t_2: 1856 年 9 月には天京事変が発生，楊秀清，韋昌輝，秦日綱が粛清された．
> <dataset type = bc(dev.) , id = 321 , label =N>

このペアの文字オーバーラップ率 ($cor(t_1, t_2)$) は 0.89 であるが，t_2 に出現する「9」という数字は，t_1 には出現しない．

数字の抽出では，t_2 から，出現するアラビア数字 (半角のみ) を抽出する．ただし，直後に「つ」が存在する場合は抽出しない．抽出した数字 n の照合には，以下の条件を用いる．

(a) n と同じ数字が t_1 に出現する
(b) n を漢数字に変換したものが t_1 に出現する

6. MC サブシステム

MC サブタスクを解くシステムを，前節で述べた BC サブシステムに，新たに導入する矛盾検出モジュールを組み合わせることで実装した．

6.1 システム概要

作成したシステムの概要を図 5 に示す．このシステムは，以下の 3 段階のステップで意味関係の認識を行う．

ステップ 1: 矛盾検出モジュールにおいて，4 つの条件のいずれかに当てはまるテキストペアを **C** と判定する．

ステップ 2: BC サブシステム (含意認識システム) を用いて，t_1 が t_2 を含意するかどうか調べる．含意関係が成立しない場合，**I**(無関係) と判定する．

ステップ 3: BC サブシステムを用いて，t_2 が t_1 を含意するかどうか調べる．含意関係が成立しない場合は，**F**(片方向含意) と判定し，成立する場合は，**B**(双方向含意) と判定する．

以下では，新たに導入した矛盾検出モジュールについて説明する．

6.2 矛盾検出モジュール

MC サブタスクのポイントは，いかにして矛盾 (**C**) を検出するかにある．後の 7 節で示すように，矛盾の自動検出は，片方向含意 (**B**) や双方向含意 (**B**) の自動認識に比べ，かなり難しい．そのため，矛盾検出の条件には，原則として，片方向含意や双方向含意の認定条件と重ならないようにする方針をとった．このような方針を採用することにより，矛盾検出の導入が，含意の認識に悪影響を与えることを防ぐことができる．

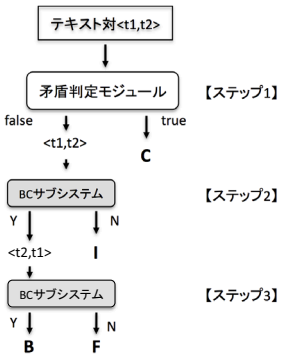

図 5 MC サブシステム概要

矛盾するテキストペアは，つぎのような性質を持つ傾向がある．

(1) 表層類似度が高い．かつ，双方向で表層類似度が高い場合が多い (矛盾は，双方向で成立する関係である)．
(2) 表層の差異が，意味的に大きな差異をもつことが多い．

これらの点を考慮し，開発用データに基づき，矛盾検出条件として次の 4 つの条件を設定した．

$$\textbf{(c1)} \begin{cases} cor(t_1, t_2) \geq 0.69 \quad \& \\ overlap\_ratio(\boldsymbol{C^2}; t_1, t_2) > 0.65 \quad \& \\ Num\_mismatch(t_1, t_2) = 1 \end{cases}$$

$$\textbf{(c2)} \begin{cases} 0.69 > cor(t_1, t_2) > 0.59 \quad \& \\ overlap\_ratio_B(\boldsymbol{C^1}; t1, t2) > 0.5 \end{cases}$$

$$\textbf{(c3)} \begin{cases} 0.69 > cor(t_1, t_2) > 0.5 \quad \& \\ ht\_ratio(t_1, t_2) > 0.4 \end{cases}$$

$$\textbf{(c4)} \begin{cases} cor(t1, t2) > 0.69 \quad \& \\ ht\_ratio(t_1, t_2) > 0.75 \end{cases}$$

これらの条件では，いずれも文字オーバーラップ率 (cor) を使用している．

最初の条件 (c1) では，文字オーバーラップ率に，5.3.2 節で説明した数字の不整合 ($Num\_mistach$) と，文字 bigram のオーバーラップ率 $overlap\_ratio(\boldsymbol{C^2}; t_1, t_2)$ を組み合わせている．後者の値が高い場合，t_1 と t_2 は，文字列としてかなり似ていることになる．それにも関わらず，数字に不整合がある場合を，矛盾とみなす．この条件により正しく矛盾と判定できる例を，以下に示す．

> t_1: 地獄の天使は，1967 年 8 月 18 日に公開されたアメリカ合衆国の映画である．
> t_2: 地獄の天使は 1930 年の映画である．
> <dataset type = mc(dev.) , id = 537 , label =C>

2 番目の条件 (c2) では，文字オーバーラップ率と，双方

向の文字オーバーラップ率とを組み合わせている．この条件により正しく矛盾と判定できる例を，以下に示す．

t_1: 留萌市は，北海道留萌管内にある市で，留萌振興局所在地で，留萌管内の中心都市である．
t_2: 留萌振興局は北海道の振興局のひとつで振興局所在地は新留萌市にある．
<dataset type = mc(dev.) , id = 447 , label =C>

残り 2 つの条件では，$ht\_ratio(t_1, t_2)$ という関数を用いる．この関数は，次のように定義される．

まず，テキストペアに対し，共通する先頭部と末尾部を切り離し，それぞれを以下のように 3 分割する．

t_1: $[h_1 h_2 \dots h_i][d_1 d_2 \dots d_k][z_1 z_2 \dots z_j]$
t_2: $[h_1 h_2 \dots h_i][d'_1 d'_2 \dots d'_l][z_1 z_2 \dots z_j]$

ここで，$[h_1 h_2 \dots h_i]$ は，t_1 と t_2 で先頭から最も長く一致する文字列であり，i が先頭から一致する最大の長さとなる．$[z_1 z_2 \dots z_j]$ は，t_1 と t_2 で末尾から最も長く一致する文字列であり，j が末尾から一致する最大の長さとなる．

このような分割に基づき，$ht\_ratio(t_1, t_2)$ を以下の式で定義する．

$$ht\_ratio(t_1, t_2) = \frac{2(i+j)}{(i+k+j)+(i+l+j)} \qquad (6)$$

この値は，2 つのテキストの先頭および末尾でそれぞれ共通する部分が，テキスト全体のどれぐらいの割合を占めるか，ということを表す．先に示した $overlap\_ratio$ が出現位置や順序を考慮しないのに対し，この値は，文字列の先頭と末尾の共通文字列にのみ着目した値である．

開発用セットでは，この値が比較的大きな場合に矛盾となる場合が多かったため，(c3) と (c4) の条件を設定した．(c3) の条件により正しく矛盾と判定できる例を以下に示す．

t_1: 大きな政府とは，政府・行政の規模・権限を 可能な限り小さくしようとする思想または政策である．
t_2: 大きな政府とは，政府・行政の規模・権限を拡大しようとする思想または政策である．
<dataset type = mc(dev.) , id = 282 , label =C>

(c4) の条件により正しく矛盾と判定できる例を以下に示す．

t_1: サイボウズ Live は，無料招待制でサービスをしている．
t_2: サイボウズ Live は招待制ではなく，だれでも自由に閲覧できる．
<dataset type = mc(dev.) , id = 388 , label =C>

7. システムの性能と考察

本節では，RITE2 の開発用データと評価用 (フォーマルラン) データに対する本システムの性能を示し，それらの結果に対して考察する．

7.1 システムの性能

表 3 に，開発用データ (dev.) と評価用データ (form.) のそれぞれに対する，本システムの精度 (acc.) および $MacroF1$(M-F1) を示す．この表の rank は，フォーマルランにおける順位を表す[*5]．Best は，それぞれのタスクで最も性能が良かったシステム，Baseline は，主催者が用意したベースラインシステム (機械学習を用いたシステム) を意味する．この表に示すように，BC サブシステムは 42 システム中 7 位，MC サブシステムは 21 システム中 1 位の成績であった．

表 3 システム性能

data	system	BC			MC		
		acc.	M-F1	rank	acc.	M-F1	rank
dev.	本システム	85.4	85.0	-	65.2	58.8	-
form.	本システム	78.9	78.6	7/42	69.5	60.0	1/21
	Best	81.6	80.5	1/42	69.5	60.0	1/21
	Baseline	63.9	62.5	29/42	45.4	26.6	16/21
(性能低下)		-6.6	-6.4	-	+4.4	+1.2	

この表の最下段の値は，評価用データに対する性能が，開発用データに対する性能からどの程度低下したかを表す．開発用データに基づいてシステムを開発するため，評価用データに対する性能は，開発用データに対する性能より低いのが普通である．BC サブシステムでは，評価用データに対する性能の方が低いという，通常通りの結果が得られた．しかしながら，MC サブシステムは，逆に，評価用データに対する性能の方が高かった．

7.2 BC サブシステムの検討

BC サブシステムの各ステップの性能を，表 4 に示す．この表から，開発用データおよび評価用データのいずれに対しても，ステップ 2 を導入することにより，性能が向上していることがわかる．しかしながら，その向上の度合は，精度で約 1–2%程度，$MacroF1$ で 0.01–0.02 程度と，それほど大きなものではない．すなわち，2 段階の戦略は機能しているが，2 段目のヒューリスティックの効果は限定的であり，改善の余地がある．

先に述べたように，評価用データに対する BC サブシステムの性能は，開発用データに対する性能よりも低い．表 4 より，その性能低下は，それぞれのステップの性能低下の積み重ねが原因となっていることがわかる．

ステップ 1 の表層類似度による判定の性能低下は，文字オーバーラップ率のしきい値，および，しきい値付近の分類精度を向上させるための拡張が，開発用データに最適な形に調整されていることによる．そのため，この性能低下はやむを得ない．

一方，ステップ 2 のヒューリスティックには，いくつかの不備が見つかった．固有名詞のヒューリスティックでは，固有名詞の抽出および照合のいずれにおいても，失敗が存在した．固有名詞の抽出では，固有名詞ではない「古称」という名詞に「自動獲得:Wikipedia」の素性が付与さ

[*5] RITE2 では，システムの性能を測る指標として $MacroF1$ が用いられた [4].

表 4 BC サブタスクの各ステップの性能

			Y	N	total	recall	prec	F1
step1	N_1		28 / 41	298 / 260	326 / 301	80% / 73%	91% / 86%	-
step2	N_{21}		4 / 7	16 / 13	20 / 20	4% / 4%	80% / 65%	-
		N(固有名詞)	4 / 5	11 / 6	15 / 11	3% / 2%	73% / 55%	-
		N(数字)	0 / 2	5 / 7	5 / 9	1% / 2% +	100% / 78%	-
	Y_{22}		208 / 208	57 / 81	265 / 289	87% / 81%	78% / 72%	-
step1	N (= N_1)		28 / 41	298 / 260	326 / 301	80% / 73%	91% / 86%	0.86 / 0.79
	Y (= N_{21}+Y_{22})		212 / 215	73 / 94	285 / 309	88% / 84%	74% / 70%	0.81 / 0.76
	total		240 / 256	371 / 354	611 / 610	-	83% / 78%	0.83 / 0.78
step1 + step2	N (= N_1+N_{21})		32 / 48	314 / 273	346 / 321	85% / 77%	91% / 85%	0.88 / 0.81
	Y (= Y_{22})		208 / 208	57 / 81	265 / 289	87% / 81%	78% / 72%	0.82 / 0.76
	total		240 / 256	371 / 354	611 / 610	-	85% / 79%	0.85 / 0.79

各欄は，「開発用データに対する性能 / 評価用データに対する性能」を表す
「+」は性能向上（評価用データに対する性能の方がよいこと）を意味する

れていたため，固有名詞とみなしてしまった．固有名詞の照合では，「祇園」と「祇園」の表記のゆれが原因で，照合に失敗した．

数字のヒューリスティックでは，「範囲」を表す表現への対処が必要であることが判明した．以下に例を示す．

> t_1: ジョン・ケネス・ガルブレイスは 1943 年から 1948 年にかけて「フォーチュン」誌の編集者を務め，1949 年にはハーヴァード大学の経済学教授に就任した．
>
> t_2: フォーチュンは，1940 年代，ジョン・ケネス・ガルブレイスを編集員として起用した．
>
> <dataset type = bc(form.) , id = 158 , label =Y>

この例では，t_2 には「1940 年代」が，t_1 には「1943 年から 1948 年にかけて」という表現が存在する．単純な数字の照合では，これらの表現が整合することを正しく判定できない．範囲表す表現を含めた形で数字を抽出し，照合することが必要である．

7.3 MC サブシステムの検討

MC サブシステムのそれぞれのステップの性能 (recall と precision) を表 5 に示す．この表より，ステップ 2 以降の **I**，**B**，**F** の判定において，開発用データに対する性能より，評価用データに対する性能の方が高かったことがわかる．BC サブシステム (含意認識システム) を用いて行うこれらの判定が好成績だったことは，MC サブタスクの含意判定において，BC サブシステムが有効に機能したことを意味する．さらに，このことが，システム全体の性能向上をもたらしている．

一方，**C**(矛盾) の判定性能は，recall と precision のいずれも低下した．4 つの判定条件別にみると，開発用データにおいて比較的 recall が高かった (c2)，(c1)，(c4) の条件は，評価用データにおいては recall はいずれも低下し，precision も (c2) を除いて低下した．(c3) の条件は，100%の precision を保ったまま recall が向上したが，全体に対しては微々たる効果しかもたらさない．すなわち，全体として，**C** の判定に成功しているとは言い難い．

B，**F** の判定性能と比較して，**C** の判定性能の低さは際だっている．今回の結果を見る限り，矛盾の検出は，表層類似度にヒューリスティックを組み合わせるだけでは難しいと言わざるを得ない．

7.4 多段階戦略の有効性

本研究では，基本方針として多段階戦略 (3 節) を採用した．特に，BC サブシステムは，この戦略をそのまま実装した形となっている．

結果的には，BC サブシステムは，MC サブシステムの一部として威力を発揮した．我々の MC サブシステムは，フォーマルランにおいて 1 位の成績を修めたが，その好成績は，先に述べたように，BC サブシステムによってもたらされている．このことにより，今回採用した多段階戦略は，一定の成果を上げたと考えてよいだろう．

残念ながら，BC サブタスクの成績は，41 システム中 7 位であり，上位 20%に入ることはできたが，トップ 3 に入ることはできなかった．5 節で述べたように，BC サブシステムの開発には，BC サブタスクの開発用データに加え，MC サブタスクの開発用データも利用した．このことが，結果的には，MC サブタスクで好成績につながり，その一方で，BC サブタスクでそれほど好成績を残せなかった原因となったと考えられる．

謝辞 本研究では RITE2 ワークショップにおいて配布されるデータを使用してシステムの開発，評価を行った．RITE2 の運営に携った皆様に深謝します．

参考文献

[1] Haim, R. B., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge, *In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment* (2006).

[2] Pham, Q. N. M., Nguyen, L. M. and Shimazu, A.: A Machine Learning based Textual Entailment Recognition System of JAIST Team for NTCIR9 RITE, *Proceeding of NTCIR-9 Workshop Meeting* (2011).

表 5 MC サブシステムの各ステップの性能

			B	F	C	I	total	recall	prec
1	C		10 / 8	9 / 11	20 / 11	20 / 13	59 / 43	31% / 18%	34% / 26%
		c2	7 / 6	7 / 9	9 / 6	19 / 10	42 / 31	14% / 10%	19% / 21% +
		c1	0 / 0	1 / 1	6 / 2	1 / 2	8 / 5	9% / 3%	75% / 40%
		c4	3 / 2	1 / 1	4 / 1	0 / 1	8 / 5	6% / 2%	50% / 20%
		c3	0 / 0	0 / 0	1 / 2	0 / 0	1 / 2	1% / 3% +	100% / 100%
2	I		11 / 9	31 / 40	21 / 31	147 / 170	219 / 241	76% / 80% +	67% / 71% +
3	B		48 / 44	16 / 7	3 / 2	7 / 8	74 / 61	58% / 63% +	65% / 72% +
	F		14 / 9	142 / 156	21 / 17	19 / 21	196 / 203	69% / 76% +	72% / 77% +

各欄は，「開発用データに対する性能 / 評価用データに対する性能」を表す
「+」は性能向上（評価用データに対する性能の方がよいこと）を意味する

[3] Shima, H., Kanayama, H., Lee, C.-W., Lin, C.-J., Mitamura, T., Miyao, Y., Shi, S. and Takeda, K.: Overview of NTCIR-9 RITE: Recognizing Inference in TExt, *Proceedings of NTCIR-9 Workshop Meeting* (2011).

[4] Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.-W., Lin, C.-J., Shi, S., Mitamura, T., Kando, N., Shima, H. and Takeda, K.: Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Workshop, *Proceedings of NTCIR-10 Workshop Meeting* (2013).

[5] 永田昌明，藤田早苗，平　博順：汎用的な意味解析技術への挑戦，技術報告，NTT コミュニケーション科学基礎研究所 (2008).

[6] 宇高邦弘，山本和英：含意要因となるテキスト中の表現と仮説の対を用いたテキスト含意認識，言語処理学会 第 18 回年次大会 発表論文集，pp. 435–438 (2012).