含意関係認識のための機械学習と全文検索

Machine Learning and Full-text Search for the Recognizing Textual Entailment

石　井　　愛，宮　下　　洋

要　約　二つの文の間に含意関係が成り立つかどうかを判別する含意関係認識は，より高度な自然言語処理を実現する技術として期待されている．我々は，2014 年度に開催された含意関係認識技術を評価する国際ワークショップ NTCIR RITE-VAL に参加し，大学入試センター試験社会科を題材とした二つのタスクにおいて，最上位の成績を収めた．本稿では，その取り組みにおける機械学習と全文検索手法を解説し，検証結果から有効性を考察する．複数の手法を評価検証し，対象とするデータの性質に合った有効な手法を見出したことが，好成績につながった．社会科のような特定のドメインにおいて，自然言語処理の精度を高めるための手法や検証方法は，専門性が高い分野で広く応用できると考えている．

Abstract　The recognizing textual entailment (RTE) that determines whether the implication relation holds between the two sentences is crucially important to establish an advanced natural language processing (NLP). We participated in the Recognizing Inference in Text and Validation (RITE-VAL) task of the NTCIR-11 Workshop, and our results for two tasks that are taken from social studies of national center test for university were the first among the other submissions. In this paper, we describe the approach adopted by our RTE system, which includes machine learning and full-text search, and discuss the effects of these methods by the experimental results. Effective methods were found out through experiments and evaluations of multiple strategies led to the outcome. We believe that our strategies and experimental evaluation methods for improving NLP of specific domains, such as social studies are widely applicable to the fields that requires high expertise.

1. は　じ　め　に

人と機械の協働を目指した取り組みの一環として，両者のインターフェースとなる自然言語処理の研究がさかんに行われている．そのなかで，二つの文の間に含意関係が成り立つかどうかを判別する含意関係認識は，情報検索や質問応答をはじめとする幅広い応用分野において，より高度な自然言語処理を実現する技術として期待されている．含意関係とは，二つの文 t1 と t2 が与えられたとき，t1 が正しいとしたら t2 も正しいという関係のことであり，その関係が成り立つかどうかを判定するタスクを含意関係認識という．我々は，文章の意味解析についての取り組みの一環として，2014 年度に国立情報学研究所が主催した国際ワークショップ NTCIR[*1] の含意関係認識の技術を競う RITE-VAL タスク[1]に参加した．そのタスクにおいて，文から抽出した特徴量に基づく機械学習を用いた含意関係認識システムと，Apache Solr[*2] を用いた全文検索システムを構築し，最上位の成績を収めた[2]．本稿では，RITE-VAL タスクでの取り組みにおける機械学習と全文検索手法を解説し，検証結果から有効性を考察する．

以降，2 章では RITE-VAL タスクの概要と構築したシステムのアプローチについて述べる．3 章では機械学習の手法，4 章では全文検索の手法について説明し，それぞれの章で検証結果

から手法の有効性を考察する．最後に5章でまとめを述べる．

2. 含意関係認識システムの概要

本章では，NTCIR RITE-VAL タスクの概要を説明するとともに，我々が開発した含意関係認識システムのアプローチについて述べる．

2.1 NTCIR RITE-VAL の概要

NTCIR は，国立情報学研究所が主催する情報検索に関わるテキスト処理技術の評価型ワークショップである．2014年に開催されたNTCIR-11のRITE-VALタスクは，大学入試センター試験（以降，センター試験）社会科科目の正誤を問う設問を題材とした含意関係認識の技術を競うタスクである．

RITE-VAL タスクには，図1に示すように，System-Validation（SV）と Fact-Validation（FV）という二つのサブタスクがある．SV タスクでは，真となる文 t1 と，含意関係を判定する対象の文 t2 が与えられる．FV タスクは，より現実の設定に近づけたタスクであり，t1 が明示的に与えられず，教科書データや Wikipedia のテキスト集合から t1 を抽出し，t2 との含意関係を判定する．

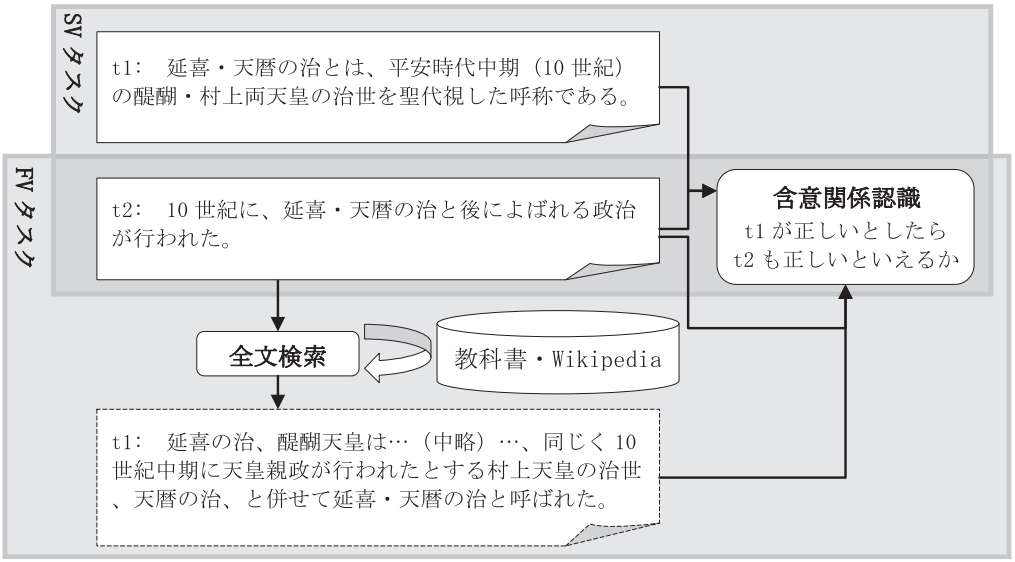

図1 SV タスクと FV タスクの概要

評価型ワークショップでは，まず正解が付与された開発用データセットが配布され，その後正解が隠された評価用データセットが配布される．その隠された正解の予測結果を提出することで，システムを評価するフォーマルランが行われる．機械学習を用いたシステムを開発する場合，開発用データセットを用いて学習を行い，評価用データセットの解答を予測する．

SV タスクおよび FV タスクの開発用データセットの例をそれぞれ図2，図3に示す．これらの例は開発用データセットであるため，「label="Y"」（含意あり）もしくは「label="N"」（含意なし）といった正解ラベルが付与されている．

```
<?xml version="1.0" encoding="UTF-8"?>
<dataset type="bc">
  <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>
  <pair id="2" label="Y">
    <t1>伊坂幸太郎は直木賞候補になった 2003 年の『重力ピエロ』で一般読者に広く認知されるようになった。</t1>
    <t2>『重力ピエロ』は伊坂幸太郎による小説で直木賞候補作品だった。</t2>
  </pair>
    :
</dataset>
```

図 2　SV タスクの開発用 XML データの例

```
<?xml version="1.0" encoding="UTF-8"?>
<dataset>
  <pair id="1" label="Y">
    <t2>パルテノン神殿の建つ丘は，アクロポリスと呼ばれている。</t2>
  </pair>
  <pair id="2" label="N">
    <t2>パルテノン神殿は，ヘレニズム文化の影響下で建設された。</t2>
  </pair>
   :
</dataset>
```

図 3　FV タスクの開発用 XML データの例

RITE-VAL タスクの成績は Accuracy（正答率）と MacroF1 によって評価され，順位付けには MacroF1 が用いられる．Accuracy および MacroF1 は以下の式で定義される．

$$Accuracy = \frac{N_{correct}}{N}$$

$$macroF1 = \frac{1}{2}\sum_{c\in\{Y,N\}} F1_c = \frac{1}{2}\sum_{c\in\{Y,N\}} \frac{2\times Prec._c \times Rec._c}{Prec._c + Rec._c}$$

ここで，N は全ての問題数，$N_{correct}$ は全ての問題のうちの正解数である．c は Y と N の 2 クラスであり，$Prec.c$ と $Rec.c$ はクラス c の Precision（適合率）と Recall（再現率）の値である．$N.c_{predicted}$ をシステムがクラス c と予測した数，$N.c_{correct}$ をクラス c と予測した中での正解数，$N.c_{target}$ は解答がクラス c の問題数とすると，$Prec.c$ と $Rec.c$ は以下の式で定義される．

$$Prec.c = \frac{N.c_{correct}}{N.c_{predicted}} \quad Rec.c = \frac{N.c_{correct}}{N.c_{target}}$$

RITE-VAL タスクの題材であるセンター試験では，四つの選択肢の中から一つの正解を選ぶ問題が多いため，正解が N である問題が多い．全て N と回答しても Accuracy が高くなる可能性があるため，Y と N クラスの F1 の平均値である MacroF1 によって順位付けが行われる．

2. 2　含意関係認識のアプローチ

2013 年に開催された同様のタスク（RITE-2）[3]において，上位の成績をおさめた Tian ら[4]のアプローチのうち浅い解析手法は，t1，t2 から，文の中で鍵となるような重要語とそれ以外の内容を表す内容語を抽出し，t2 の単語が t1 に含まれる比率などを数値化した特徴量を用いたロジスティック回帰[*3]により，Y，N のクラスを判定する手法であった．我々はその手法をベースとし，重要語の抽出および単語の対応付けの精度の改善を試みた．また，新たな特徴量を複数採用し，有効性の検証を行った．

また，FV タスクでは，与えられた t2 を用いて教科書や Wikipedia から t2 の根拠となる t1 を抽出するための，全文検索が必要となる．インターネット検索などの一般的な全文検索では，Web ページであれば 1 ページの単位，文書ファイルであれば 1 ファイル単位というように，1 文書を検索単位としている．一方，センター試験社会科の選択肢の一文である t2 の根拠となる t1 は 1 文書よりも局所的な箇所であり，かつ，t2 に含まれる単語群の含有率が高い箇所であると考えられ，一般的な全文検索とは目的が異なる．そのような t1 を抽出する全文検索を実現するため，複数の検索手法の評価検証に取り組んだ．

3.　機械学習による含意関係認識

本章では，含意関係認識のための機械学習手法について説明し，検証結果から有効性を考察する．3. 1 節では含意関係認識のための機械学習手法，3. 2 節では機械学習で使用する特徴量抽出について説明する．3. 3 節では特徴量抽出で算出する七つの特徴量の有効性を検証結果から考察し，3. 4 節では特徴量抽出における固有表現の抽出と単語の対応付けについて，エラー解析結果から考察する．

3. 1　機械学習手法

機械学習による含意関係認識は図 4 に示すように，大きく学習フェーズと判定フェーズに分かれる．学習フェーズでは，まず，開発用データセットの t1，t2 それぞれから数量/時間表現，固有表現，内容語を抽出し特徴量を算出する．次に，開発用データセットに付与されている Y または N の正解ラベルを用いて学習し，判別式を作成する．判定フェーズでは，同様に特徴量を算出し，その特徴量を判別式に入力することで含意関係を判定する．学習モデルとしてはロジスティック回帰を用い，判別式の閾値を 0.5 として Y，N の 2 値に分類する．

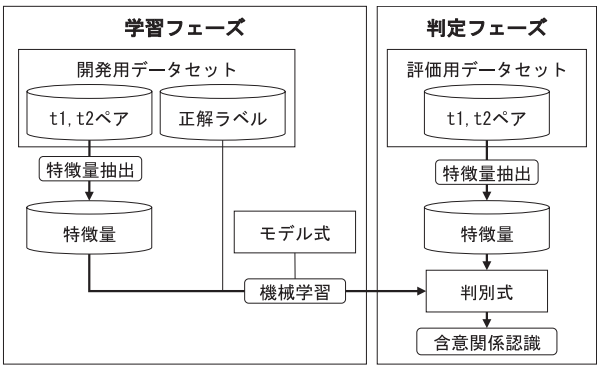

図 4　機械学習による含意関係認識の処理の流れ

3. 2　特徴量抽出

特徴量の抽出では，図 5 に示すように「重要語と内容語の抽出」，「単語の対応付け」，「特徴量の算出」という三つの処理を行う．以降の項では，特徴量抽出のそれぞれの処理について説明する．

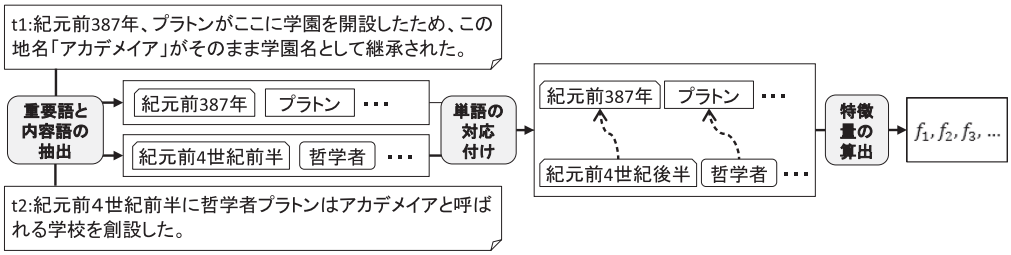

図 5　特徴量抽出の流れ

3. 2. 1　重要語と内容語の抽出

t1 と t2 それぞれから，文の中で鍵となる重要語として数量/時間表現および固有表現，それ以外の単語を内容語として抽出する．Tian らの手法では，Wikipedia タイトルから日本語語彙体系[5]の普通名詞を除外したものを固有表現とし，時間・数量表現抽出ツールである NormalizeNumexp[*4] を使用して数量/時間表現を抽出している．その手法を参考に，表 1 に示す方針で重要語と内容語の抽出を行う．ただし，Tian らの手法では，文節ごとに左から最長一致する方法で単語を抽出するため，文節の途中からはじまる固有表現や，文節をまたがる固有表現が抽出されない．そのような問題に対応するため，本手法では，まず文節よりも細かい形態素の単位に文を分割し，次に各形態素に対し，その形態素からはじまる固有表現の抽出を試みる．これにより，文節をまたがる固有表現も抽出できる．文の分割には KNP[*5] を用いる．

表 1　重要語と内容語の抽出方法

区分	抽出方法
重要語 （数量/時間表現）	NormalizeNumexp にて抽出する．
重要語 （固有表現）	以下の条件を満たす単語を固有表現として，最長一致で抽出する． ・Wikipedia タイトルから日本語語彙体系の普通名詞を除外したもの ・日本語語彙体系の固有名詞 ・品詞が固有名詞の形態素 ただし，1 文字の単語，ひらがなだけの単語は除く．
内容語	重要語以外の，以下の条件を満たす単語を内容語として，最長一致で抽出する． ・日本語語彙体系の普通名詞 ・品詞が非自立詞，接尾詞以外の形態素

固有表現は，表 1 に示したとおり，主に Wikipedia のタイトルから抽出する．ただし，Wikipedia のタイトルには「日本の歴史」，「空港一覧」，「モンゴル（曖昧さ回避）」など固有表現としてふさわしくない単語が多く含まれる．そのため，関口[6]の「日本語 Wikipedia にお

ける不適切なエントリ削除のためのルール」を参考に，不要な単語を除外するパターンリストを作成し，適用した．また，1 文字の単語や，ひらがなだけの単語を取り除くなどの調整も合わせて実施した．

3. 2. 2　単語の対応付け

単語の対応付けでは，単語の文字列が完全に一致する場合だけでなく，意味が同じかどうかを考慮する必要がある．

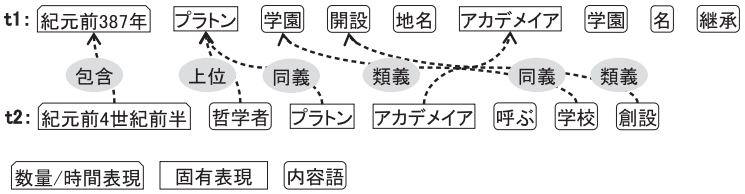

図 6　重要語と内容語の意味に基づく単語の対応関係

図 6 は，t2 の単語が t1 内のどの単語に対応するかを示したものであり，点線で示した単語同士は，この文中では意味が同一とみなしたい単語である．「紀元前 4 世紀前半」と「紀元前 387 年」は時間的な意味で包含関係にある．時間の包含関係に対応するため，NormalizeNumexp によって規格化された情報を用いて，包含関係を判定するモジュールを作成して対応した．また，「学校」と「学園」，「創設」と「開設」は，ほとんど同じ意味であり，同義・類義関係にある．同義・類義関係の判定には，Wikipedia リダイレクト，日本語 WordNet[7] の同一概念の単語群，日本語語彙体系の見出し語の異表記語を用いた同義・類義語辞書を作成し，対応付けを行った．さらに，辞書の語彙の不足を考慮して，一致の判定にはレーベンシュタイン距離[*6] を取り入れた．

また，プラトンが哲学者である，という知識があれば「哲学者」は「プラトン」の上位語として，対応付けることができる．上位下位関係[*7] の判定には，Wikipedia から自動的に上位下位語を抽出する上位下位関係抽出ツール[8] の出力を利用して上位下位語辞書を作成し，対応付けを行った．

3. 2. 3　特徴量の算出

t1 と t2 の単語の対応関係に基づき，特徴量 $f_1, \ldots, f_7$ を表 2 に示す方法で算出する．Tian らの手法で有効性が示されている，数量/時間表現および固有表現の一致と，内容語一致率を考慮する $f_1 \sim f_3$ の特徴量に加えて，本手法では $f_4 \sim f_7$ の四つの特徴量を作成した．これらの特徴量を用いた機械学習により，t1 と t2 の含意関係（ラベル Y/N）を判定する．

表 2　特徴量

名称	表記	説明
数量/時間表現一致	f_1	t2 のすべての数量/時間表現に対応する単語が t1 に存在する場合は 0.9，それ以外は 0.1.
固有表現一致	f_2	t2 のすべての固有表現に対応する単語が t1 に存在する場合は 0.9，それ以外は 0.1.
内容語一致率	f_3	t2 の内容語のうち対応する単語が t1 に存在する比率.
内容語の先頭文字一致率	f_4	t2 の内容語のうち t1 に先頭の文字列が一致する単語が存在する比率. 例）「米国」と「米議会」を一致とみなす.
word2vec[*8] コサイン距離	f_5	t2 の内容語と t1 の単語間で最大となる word2vec のコサイン距離．t2 には複数の内容語が含まれるため平均を取る．word2vec の学習には日本語 Wikipedia 全体を使用する．同義・類義語辞書に登録されていない意味的に近い意味の単語の一致を検知する.
排他語の有無	f_6	t1 の単語と対応がない t2 の内容語の中で，t1 に排他語を持つ場合は 1.0，それ以外は 0.1. 排他語は，日本語 WordNet 上で特定の属性の配下に存在する単語同士とする. 例）spectral_colour（色）の下位概念同士の「青」と「赤」を排他語とする.
内容語不一致率	f_7	t1 の単語と対応がない t2 の内容語の比率. t1，t2 がまったく無関係であることを検知する.

3. 3　検証

前節で示した特徴量 $f_1, \ldots, f_7$ について，特徴量のサブセットから構成されるモデルの Macro-F1 および Accuracy を比較することにより，特徴量の有効性を検証し，その結果について述べる.

3. 3. 1　対象データセット

SV タスクのデータセットの統計量を表 3 に示す.

表 3　SV タスクのデータセット統計量

データセット	Y	N	合計
開発用	450（40.1%）	671（59.9%）	1,121
評価用	339（24.6%）	1,040（75.4%）	1,379
開発 + 評価用	789（31.6%）	1,711（68.4%）	2,500

表 3 に示したとおり，開発用と評価用のデータセットでは Y/N 比が大きく異なっていた．また，対象とするデータは，評価用データセットがセンター試験社会科（歴史，地理および公民）の正誤問題の選択肢を 1 文ずつに加工したものであるのに対し，開発用データセットはセ

ンター試験社会科の問題と，Wikipedia から抽出した様々な話題を題材とした問題によって構成されており，性質が異なるものであった．

3. 3. 2　検証方法

本検証では，特徴量 $f_1, \ldots, f_7$ の一部のみを使用したモデルを複数作成し，それぞれのモデルを MacroF1 と Accuracy の値により評価する．評価対象とするモデルは，以下の式で示す損失関数（loss）を指標とし，ステップワイズ法[*9] によりモデルで使用する特徴量を一つずつ増加させた場合に出現するモデルとする．

$$\text{loss} = 1 - \text{MacroF1}$$

それぞれのモデルにおける MacroF1 と Accuracy は，K 分割交差検証（K 分割クロスバリデーション）[*10] を使用して算出する．その際，分割数を 50 とした．

3. 3. 3　検証結果

検証結果を表 4 に示し，また，表 4 の MacroF1 の推移を図 7，Accuracy の推移を図 8 に示す．ただし，表 4 の「評価」データセットにおける結果は，開発用データセットを用いて学習した判別式を評価用データに適用した際の値である．

図 7 から，特徴量を f_3，f_1，f_2 の三つまで追加すると，すべてのデータセットにおいて MacroF1 の値が向上することがわかる．このことから，今回使用したデータセットにおいては「固有表現の一致（f_1）」，「数量/時間表現の一致（f_2）」，「内容語の一致率（f_3）」の三つの特徴量が安定して有効であることがわかった．一方，その他の特徴量 f_4，f_5，f_6，f_7 については，対象とするデータセットによって，MacroF1 が向上する場合と低下する場合に分かれた．このことから，特徴量 f_4，f_5，f_6，f_7 の有効性は，データセットの性質に依存するものと考えられる．

表 4　特徴量選択による評価結果

使用する特徴量	特徴量数	開発		評価		開発 + 評価	
		MacroF1	Accuracy	MacroF1	Accuracy	MacroF1	Accuracy
—	0	37.18	59.85	42.99	75.42	40.54	68.44
f_3	1	72.44	75.20	67.34	75.78	67.89	76.28
f_3，f_1	2	73.84	76.44	68.13	77.23	68.93	76.84
f_3，f_1，f_2	3	74.60	76.82	69.16	77.01	71.24	77.74
f_3，f_1，f_2，f_6	4	74.71	76.67	68.77	76.79	70.99	77.50
f_3，f_1，f_2，f_6，f_4	5	74.67	76.72	68.52	76.79	70.96	77.55
f_3，f_1，f_2，f_6，f_4，f_5	6	74.49	76.62	69.19	77.52	71.34	77.56
f_3，f_1，f_2，f_6，f_4，f_5，f_7	7	74.27	76.50	69.59	77.81	71.43	77.42

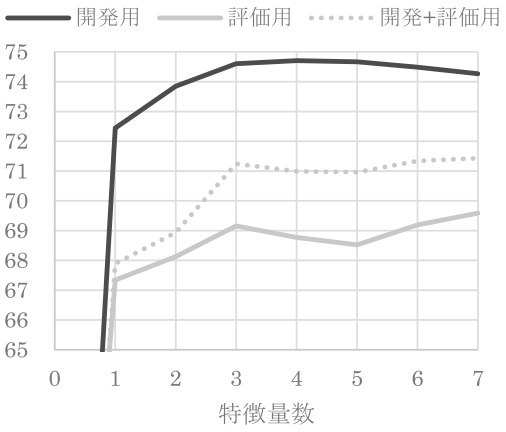

図 7　特徴量選択による MacroF1 の推移

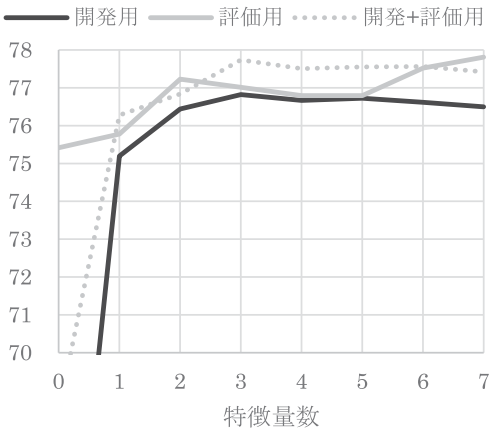

図 8　特徴量選択による Accuracy の推移

3. 4　考察

3. 1 節で示した特徴量抽出の重要な処理である，重要語と内容語の抽出における固有表現の抽出，および単語の対応付けについて，エラー分析の結果から考察を述べる．

3. 4. 1　固有表現の抽出

重要語のうち，固有表現の抽出については，Wikipedia のタイトルリストから日本語語彙体系の普通名詞を除外する手法と，形態素ごとに抽出を試みる手法が良く働き，精度を大きく向上させることができた．しかし，エラー解析の結果，本来抽出するべき固有表現のうち，Wikipedia タイトルに含まれない単語や，除外対象とした日本語語彙体系の普通名詞に含まれる単語も少なからず存在した．また本手法では，誤抽出を少なくするために 1 文字の単語は重要語から除外する方針としたが，中国の歴史上の王朝名である「清」や「明」などに対応することができず，中国史の問題の正解率が下がる原因となっていた．センター試験の社会科のような対象とする範囲が限られるドメインでは，ある程度人手で辞書を整備することは現実的である．精度の高い固有表現辞書を用意することができれば，システム全体の精度がより向上すると考える．

3. 4. 2　単語の対応付け

類義語，上位語，レーベンシュタイン距離による単語の対応付けは，それぞれ実施することでシステム全体の精度が向上し，特に上位語による対応付けはシステム全体の精度に大きく貢献した．それぞれの単語の対応付けについて，エラー解析の結果から適切な対応付けの例と誤った対応付けの例を表 5 に示す．

類義語については悪い例が比較的少なかったものの，日本語 WordNet で対象の単語が属する複数の概念すべてを結びつけたことを原因とする誤りや，Wikipedia のリダイレクトから作成した辞書の誤りによる，対応付けが必要のない単語同士が類義語と判定される例があった．上位語の対応付けについては，上位下位関係抽出ツールの判別誤りに起因する，誤った対応付けの例があった．同義類義関係および上位下位関係のより精緻な対応付けのためには，辞書を作成した後に誤ったものを取り除くような対応が必要となる．

レーベンシュタイン距離については，誤った例に示したように，1 文字異なることで意味がまったく異なる単語が対応付けられていた例があった．特に固有表現を対象としたレーベンシュタイン距離の適用には，閾値や適用する条件を再検討する必要がある．

表 5　単語の対応付けの例

区分	適切な対応付けの例	誤った対応付けの例
類義語	UNESCO ≒国連，増加≒上昇， 醍醐天皇≒延喜	直接≒間接，協定≒案配
上位語	連載小説⊂作品，ウジェーヌ・ドラクロワ⊂画家，著作権⊂知的財産権	衆議院⊂国会議員，銀行⊂各国， 搾取⊂企業
レーベンシュタイン距離	ユーゴスラビア≒ユーゴズラヴィア（距離：3），ゴードン・ブラウン内閣≒ゴードン・ブラウン改造内閣（距離：2）	アメリカ≒アフリカ（距離：1）， ルイ 15 世≒ルイ 14 世（距離：1）

4.　全文検索

本章では，FV タスクにおいて，与えられた t2 の根拠となる適切な t1 を導くための全文検索手法について説明し，検証結果から検索手法の有効性を考察する．4. 1 節では適切な t1 を導くため四つの観点から検討した検索手法について説明し，4. 2 節では各検索手法の有効性の検証とその結果について述べる．4. 3 節では 4. 2 節の検証結果を基に各検索手法の有効性について考察する．

4. 1　全文検索手法

全文検索エンジンの検索精度の向上には，検索インデックスの作成，単語辞書の活用，検索クエリ生成，スコア計算といった大きく四つの観点からの工夫が考えられる．t2 から適切な t1 を導くための検索は，一般的な文書検索とは目的が異なるため，四つの観点それぞれで有効な手法を検討した．

全文検索エンジンは Solr を用い，あらかじめ教科書と Wikipedia のデータから検索インデックスを作成した上で，検索結果においてスコアが 1 番高い箇所を t1 とする．

4.1.1　検索インデックスの単位

適切な t1 は，検索クエリで指定される t2 の単語群が近接して存在する箇所であると仮定した．Web ページや文書ファイルを対象とした検索では，一つの文書内で単語が分散して存在していて近接度が低くても，単語が多く含まれていればスコアは高くなる．この現象を防ぐための単語の近接度を検索スコアに反映する検索手法も存在するが，全文検索システムには実装されていないものが多い．そのため，検索インデックスの単位を適切に調整することを検討した．

狩野[9]はセンター試験世界史を対象とした全文検索において，教科書データの節・小節・段落を，それぞれ検索単位として検証し，各年度の試験の成績を報告している．その結果では，試験の年度によって結果が異なるものの，段落単位が比較的良い成績であった．そのため，本試行では，教科書および Wikipedia を段落単位に区切ったデータを，検索インデックスの 1 文書とした．

4.1.2　単語辞書の活用

全文検索では，新しい単語や専門用語に対応して形態素解析を行うためのユーザ辞書や，同義語や類義語に対応するための類義語辞書を適用することができる．

FV タスクのテキスト集合である Wikipedia と教科書データの記述中において，正誤判定の証拠となる部分は各データ内で一度しか現れないことが多い．たとえば一冊の教科書の中に，同じ歴史的なイベントの記述が重複して現れることは稀だと推測できる．それを考慮し，形態素解析用のユーザ辞書は検索漏れが発生しやすくなるため適用せず，類義語辞書の適用を試行した．類義語辞書は Wikipedia のリダイレクト情報から自動的に作成した．

4.1.3　検索クエリ生成

t2 から抽出する重要語と内容語のなかでも，重要語が多く含まれる t1 が適切であるという仮説から，重要語の優先度を高くする検索クエリを試行した．検索クエリでは，単語ごとにその単語が含まれている場合のスコアを何倍にするといった，重み付けを指定することができる．重み付けの倍率をいくつか試行した上で，重要語を 5 倍に重み付けをした検索クエリを試行した．

4.1.4　スコア計算

検索エンジンのスコア計算では，文書内の単語の出現数である用語頻度（TF）と単語の重要度としての逆文書頻度（IDF）の二つの指標に基づいて計算される TF-IDF という重み付けが一般的である．IDF は，全体の文書数のうち，対象の単語が含まれる文書数の割合（DF）の逆数から計算される．また，TF-IDF において文書が長いほどスコアが高くなる傾向を調整するための，文書長によって重みを正規化する文書長正規化も広く採用されている．

Solr のデフォルトのスコア計算[*11]は，TF-IDF の変形と文書長正規化を考慮したものであり，検索クエリの単語 t の文書 d におけるスコア $w_{t,d}$ は，以下の式で定義されている．

$$w_{t,d} = tf_{t,d}^{\frac{1}{2}} \cdot idf_t^2 \cdot boost_t \cdot boost_f \cdot lengthNorm_d$$

ここで，$boost_t$ は検索クエリで指定する単語 t に対する重み，$boost_f$ は検索フィールド[*12] に指定する重み，$lengthNorm_d$ は文書長正規化の係数である．単語数が $numTerms_d$ の文書 d における $lengthNorm_d$ は，以下の式で定義されている．

$$lengthNorm_d = \frac{1}{\sqrt{numTerms_d}}$$

検証では，t2 内の特定の単語の出現頻度よりも，t2 の単語群の含有率が高い箇所が t1 として適切であるという仮説から，$tf_{t,d} = 1$ と固定することで TF を無効化するスコア計算を試行した．また，文書長正規化がスコア計算に含まれていると，t2 に含まれる単語群の含有率が高い箇所よりも，単語数が少ない箇所のスコアが高くなる可能性がある．そのため，$lengthNorm_d = 1$ と固定することで文書長正規化を無効化するスコア計算を試行した．

4. 2　検証

前節で示した全文検索手法について，評価実験の内容および結果を述べる．

4. 2. 1　対象データセット・リソース

FV タスクのデータセットの統計量を表 6 に示す．対象とするデータは，センター試験社会科（歴史，地理および公民）の正誤問題の選択肢を 1 文ずつに加工したものである．

表 6　FV タスクのデータセット統計量

データセット	Y	N	合計
開発用	383（40.0%）	575（60.0%）	958
評価用	208（40.5%）	306（59.5%）	514
開発 + 評価用	591（40.1%）	881（59.9%）	1,472

また，検索対象とした教科書データおよび Wikipedia の統計量を表 7 に示す．

表 7　検索対象データの統計量

検索対象データ	データ件数（段落単位）
教科書データ	13,663
Wikipedia	9,133,199
合計	9,146,862

4. 2. 2　検証方法

検証する検索手法の各試行と，その実装方法を表 8 に示す．検証では，機械学習手法は固定し，検索手法の検証項目のみを変更した実行結果を評価する．ベースラインは，検索インデックス単位をページ単位，検索クエリを t2 の文をそのまま指定，スコア計算は Solr のデフォルトを採用する手法とし，各検索手法の試行結果と比較する．「4. 段落 + 重要語重み」の試行に

ついては，「5. 段落＋TF 無効」と「6. 段落＋LN 無効」のそれぞれを組み合わせた試行と，三つを組み合わせた試行を評価する．

表 8　検索手法の試行と実装方法

試行 ID	インデックス	検索クエリ	スコア計算	実装方法
1. ベースライン	ページ	t2 の文	Solr のデフォルト	最大 100 字と設定したスニペット[*13] を最大三つ結合した文字列を t1 とする．
2. 段落単位	段落	t2 の文	Solr のデフォルト	教科書データでは明示されている文書構造の段落を利用し，Wikipedia データでは本文を改行で区切ったデータを検索単位とする．
3. 段落＋類義語	段落	t2 の文＋検索クエリを類義語展開	Solr のデフォルト	Solr のスキーマ定義により，類義語辞書を指定し，検索クエリを類義語展開するように設定する．
4. 段落＋重要語重み	段落	重み付けした重要語＋内容語	Solr のデフォルト	重要語を 5 倍に重み付けした文字列と内容語を OR で連結した文字列を検索クエリとする．
5. 段落＋TF 無効	段落	t2 の文	TF＝1	Solr の DefaultSimilarity クラスで用語頻度（TF）を無効化する．
6. 段落＋LN 無効	段落	t2 の文	LN＝1	Solr の DefaultSimilarity クラスで文書長正規化係数（LN）を無効化する．
7. 段落＋重要語重み＋TF 無効（4＋5）	段落	重み付けした重要語＋内容語	TF＝1	4. 段落＋重要語重み，5. 段落＋TF 無効の組み合わせ．
8. 段落＋重要語重み＋LN 無効（4＋6）	段落	重み付けした重要語＋内容語	LN＝1	4. 段落＋重要語重み，6. 段落＋LN 無効の組み合わせ．
9. 段落＋重要語重み＋TF 無　効＋LN 無効（4＋5＋6）	段落	重み付けした重要語＋内容語	TF＝1，LN＝1	4. 段落＋重要語重み，5. 段落＋TF 無効，6. 段落＋LN 無効の組み合わせ．

4. 2. 3　検証結果

開発用，評価用データセットおよびデータセット全体を用いて検証した各検索試行の検証結果の MacroF1，Accuracy を表 9 に示す．評価用データセットにおける結果は，開発用データセットを用いて学習した判別式を適用した際の値であり，フォーマルラン後に配布された正解ラベルを用いた．また，開発用と評価用で MacroF1 が高くなる傾向が異なったため，データセット全体を用いた検証結果を併せて示した．開発用およびデータセット全体を用いた検証は，10 分割交差検証を用いた．

表 9　各検索試行の検証結果

試行 ID	開発		評価		全体	
	MacroF1	Accuracy	MacroF1	Accuracy	MacroF1	Accuracy
1. ベースライン	62.23	63.74	59.80	61.67	60.01	63.04
2. 段落単位	66.18	66.98	62.61	63.42	65.06	65.99
3. 段落＋類義語	65.92	66.81	62.49	63.62	64.90	65.97
4. 段落＋重要語重み	65.55	66.46	61.93	63.23	64.22	65.56
5. 段落＋TF 無効	65.28	66.70	61.64	63.04	63.97	65.38
6. 段落＋LN 無効	64.91	65.35	64.28	64.59	64.21	64.86
7.（4＋5）	65.14	66.32	58.68	61.09	62.38	64.04
8.（4＋6）	58.67	61.24	59.09	61.87	58.52	61.91
9.（4＋5＋6）	59.95	63.22	57.51	60.89	57.95	62.23

また，システムの評価指標である MacroF1 および Accuracy は，検索手法の評価指標としては機械学習後の間接的な結果である．各検索手法を用いた試行において，どのような t1 を抽出していたかを直接評価するため，正解ラベルが Y の場合と N の場合に分け，t1 における t2 内の重要語と内容語のそれぞれの含有率と，t1 の単語の個数の平均値を算出した．正解ラベルが Y の場合は根拠となる t1 が見つかるはずであり，そのような t1 は重要語と内容語の含有率が高く，N の場合はそれらが低いと考えられる．

評価用データセットを用いた各検索試行における t1 の特徴を表 10 に示す．さらに，t1 の特徴と MacroF1 との関係を図 9 に示す．図 9 の単語の含有率（Y/N）は，正解ラベルが Y の場合と N の場合それぞれの，重要語の含有率と内容語の含有率の平均値である．

表 10　t1 の単語の含有率および単語数

試行 ID	Y			N		
	重要語含有率	内容語含有率	t1 単語数	重要語含有率	内容語含有率	t1 単語数
1. ベースライン	0.76	0.76	70.15	0.68	0.70	69.70
2. 段落単位	0.78	0.72	42.73	0.61	0.66	42.15
3. 段落＋類義語	0.76	0.72	42.24	0.63	0.66	43.85
4. 段落＋重要語重み	0.85	0.69	41.13	0.76	0.63	42.01
5. 段落＋TF 無効	0.72	0.68	34.50	0.57	0.62	31.59
6. 段落＋LN 無効	0.82	0.81	106.17	0.67	0.75	114.61
7.（4＋5）	0.82	0.62	31.47	0.73	0.59	31.91
8.（4＋6）	0.87	0.79	114.12	0.75	0.73	114.50
9.（4＋5＋6）	0.86	0.81	96.02	0.77	0.75	99.94

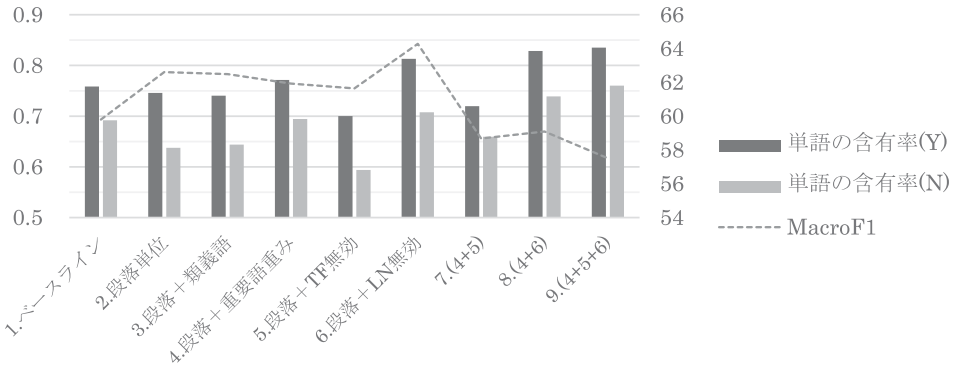

図 9　t1 の単語含有率と MacroF1 の関係

4. 3　考察

4. 2. 3 節に示した検証結果から，適切な t1 を抽出するための有効な検索手法について考察する．

表 9 の MacroF1 および Accuracy の結果では，評価用データセットでは「6. 段落＋LN 無効」，データセット全体では「2. 段落単位」がもっとも良い結果を示した．一方，表 10 を見ると，「2. 段落単位」は t1 の単語の含有率は他の試行よりも比較的低かった．図 9 を見ると，t1 の単語の含有率がもっとも高い値を示したのは「9. 段落＋重要語重み＋TF 無効＋LN 無効（4＋5＋6）」であったが，MacroF1 はもっとも低い値となった．その理由として，正解ラベルが N の場合の単語の含有率が高く，機械学習の特徴量において正解ラベルが Y の場合と N の場合の差が減少したために，機械学習結果である MacroF1 が悪化したといえる．含意関係認識に用いる適切な t1 を選択するための検索手法としては，追加の考慮が必要であることがわかった．

以降の項で，各検索手法について考察する．

4. 3. 1　検索インデックスの単位

「1. ベースライン」と「2. 段落単位」を比較した結果，すべてのデータセットにおいて MacroF1 が 3 ～ 5 ポイント高く（表 9），段落単位のインデックスが有効であることがわかった．これはページ内に散らばっている単語が含まれる箇所をつなぎ合わせるよりも，一つの段落に単語がまとまっている箇所のほうが t1 として適切であったことを示している．ただし，適切な t1 のサイズは題材とするデータセットの性質によって異なるため，題材によって検索インデックスの単位を調整する必要があると考える．

4. 3. 2　類義語辞書の適用

類義語辞書を適用した「3. 段落＋類義語」については，「2. 段落単位」と比較して，MacroF1 が若干低く（表 9），t1 の単語の含有率はほぼ同様の値（表 10）となり，有効性が観察されなかった．要因として，エラー解析の結果から，検索エンジン内部の単語の単位と類義語辞書の単語の単位が合致しないことによる，意図しない類義語展開が発生していたことがあげられる．これを防ぐためには，形態素解析用のユーザ辞書と類義語辞書の表記を揃えて整備し，両方の辞書を適用する必要がある．ただし，ユーザ辞書に適用する場合，検索漏れの対策を別途行う必要がある．

また，類義語辞書を適用しない状態においても，検索対象のうち Wikipedia では，Wikipedia の記述方法による他のページへのリンク箇所に，ページタイトルとその異表記が並べて記述されていることで，類義語に対応できていたケースがあった．そこで，その記述箇所を除去した検索インデックスを作成したところ，MacroF1 が約 2 ポイント低下した．このことから，該当箇所のページタイトルの異表記が類義語として活用されていたことがわかった．

4. 3. 3　重要語に重み付けをした検索クエリ

検索クエリで重要語に重み付けをした「4. 段落＋重要語重み」は，「2. 段落単位」と比較して，この試行で意図したとおり t1 の重要語の含有率が向上（表 10）したが，MacroF1 は低い（表 9）結果となった．MacroF1 が低い結果となった要因としては，正解ラベルが N の場合も t1 の重要語の含有率が高い値となったために，Y の場合と N の場合の機械学習の特徴量の差が減少したことが考えられる．

4. 3. 4　スコア計算

スコア計算における用語頻度（TF）を無効化した「5. 段落＋TF 無効」については，Solr のデフォルトのスコア計算である「2. 段落単位」と比較して，MacroF1（表 9）および t1 の単語の含有率（表 10）の両方が低い結果となった．本試行で検索単位とした段落はある程度の長さがあり，t1 として適切な箇所であっても，同じ単語が複数回出現する場合が多かったと考えられる．t1 の単語数（表 10）が他の試行よりも小さいことから，TF の無効化によって単語が複数回出現する箇所が選択されにくくなり，段落のなかでも短い箇所が優先されるようになったといえる．そのため，t1 の単語の含有率が低くなり，MacroF1 が低い結果になったと考えられる．

スコア計算における文書長正規化（LN）を無効化した「6. 段落＋LN 無効」については，「2. 段落単位」と比較して，意図したとおり，t1 の重要語と内容語両方の含有率が高い結果となった（表 10）．しかし，MacroF1 は評価用データセットでは一番高い結果となったものの，データセット全体では「2. 段落単位」よりも低い結果となった（表 9）．t1 の単語数（表 10）を見ると，「2. 段落単位」と比較して 2 倍以上の単語数となっており，LN の無効化によって，段落の単位のなかでも長い箇所が t1 として選択されていた．t1 が長いことで，正解ラベルが N の場合においても単語の含有率が高くなり，機械学習の特徴量において正解ラベルが Y の場合と N の場合の差が減少し，MacroF1 の改善につながらなかったと考えられる．このことから，適切な t1 を選択するためには，LN を無効化するよりも，検索単位の長さのばらつきを小さくすることが有効だと考えられる．検索単位の長さのばらつきを小さくするためには，1 文単位や 2 文単位，あるいは何単語というウィンドウサイズ（考慮する範囲）を利用する方法がある．

5. お　わ　り　に

本稿では，2014 年度に開催された NTCIR RITE-VAL タスクにおける，大学入試センター試験の社会科科目を題材とした含意関係認識についての取り組みを報告した．フォーマルランにおいては，SV タスクでは MacoroF1 69.59，Accuracy 77.81，FV タスクでは，MacroF1 61.93，Accuracy 63.23 という成績となり，両タスクにおいて参加チームの中で 1 位という結果を獲

得した．機械学習および全文検索について複数の手法を評価検証し，有効なものを見出すことができたことが好成績につながったと考えられる．

我々は本取り組みでの経験を生かし，2015 年 10 月に開催された「ロボットは東大に入れるか」プロジェクト[*14] におけるセンター試験の模試である「進研模試 総合学力マーク模試」の世界史 B に挑戦し，受験生の平均を 30 点上回る 76 点という好成績を収めることができた．センター試験世界史 B と，RITE-VAL タスクの題材であったセンター試験社会科（歴史，地理および公民）は，対象とするデータの性質に差異はあったものの，本取り組みにおける特徴量抽出手法や，検索手法およびその検証方法を生かすことができた．

そのような，社会科や世界史といった特定のドメインにおいて，自然言語処理の精度を高めるための手法や検証方法は，専門性が高い分野において広く応用できると考えている．

＊ 1　NTCIR（エンティサイル，NII Testbeds and Community for Information access Research の略）は，NII が主催する情報検索・アクセス技術の評価と性能比較の研究基盤．
Web サイト：http://ntcir.nii.ac.jp/jp/

＊ 2　Apache Solr はオープンソースの全文検索エンジン．Web サイト：
http://lucene.apache.org/solr/

＊ 3　ロジスティック回帰は，主にクラス分類に用いられる機械学習のモデルの一種．

＊ 4　NormalizeNumexp の Web サイト：
http://www.cl.ecei.tohoku.ac.jp/~katsuma/software/normalizeNumexp/

＊ 5　KNP は，京都大学にて研究開発された日本語文の構文・格・照応解析を行うシステム．
Web サイト：http://nlp.ist.i.kyoto-u.ac.jp/?KNP

＊ 6　レーベンシュタイン距離は，一方の文字列をもう一方の文字列に変換するために必要な，文字の挿入や削除，置換などの手順の最小回数．

＊ 7　上位下位関係とは，「X は Y の一種（一つ）である」といえる X と Y の関係のこと．この場合，X を下位語（下位概念），Y を上位語（上位概念）と呼ぶ．

＊ 8　word2vec は，「同じ文脈で利用される単語は，近い意味を持つ」という仮説に基づき，単語の特徴をベクトルで表現する技術．2013 年に Google 研究所が発表して以来，世界中の自然言語処理研究者・開発者の間で流行した．Web サイト：
https://code.google.com/p/word2vec/

＊ 9　ステップワイズ法は，評価値を最も改善する特徴量を増加，もしくは最も悪くなる特徴量を減少させながら特徴量のサブセットを評価する手法．評価回数を大幅に減らすことができる．

＊10　交差検証は，データセットを分割し，その一部で学習し，残りでテストを行う検証を繰り返すことにより機械学習のモデルを評価する手法．

＊11　Solr のスコア計算が記載されている TFIDFSimilarity の Web サイト：
http://lucene.apache.org/core/4_8_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

＊12　Solr 等の検索エンジンではタイトルや URL，本文といったように一つの文書を項目にわけて登録することができ，その項目を検索フィールドと呼ぶ．

＊13　スニペットは，検索結果ページで表示される検索語が含まれるテキストの断片．

＊14　「ロボットは東大に入れるか」は国立情報学研究所が主催する人工知能プロジェクト．機械のみによって東大入試レベルの統合的な人工知能を実現しようとする試みであり，大学入試をベンチマークに設定している．Web サイト：http://21robot.org/

※上記注釈に含まれる URL は 2016 年 2 月 25 日時点での存在を確認．

参考文献　［1］ S. Matsuyoshi, Y. Miyao, T. Shibata, C.-J. Lin, C.-W. Shih, Y. Watanabe and T. Mitamura, “Overview of the NTCIR-11 Recognizing Inference in Text and Validation (RITE-VAL) Task”, Proceedings of the 11th NTCIR Conference, Tokyo, Japan, 2014.

［2］ A. Ishii, H. Miyashita, M. Kobayashi and C. Hoshino, “NUL System at NTCIR RITE-VAL tasks”, Proceedings of the 11th NTCIR Conference, Tokyo, Japan, 2014.

［3］ Y. Watanabe, Y. Miyao, J. Mizuno, T. Shibata, H. Kanayama, C.-W. Lee, C.-J. Lin, S. Shi, T. Mitamura, N. Kando, H. Shima, and K. Takeda. “Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10”, In Proceedings of the 10th NTCIR Confer-

ence, 2013.
［4］ R. Tian, Y. Miyao, T. Matsuzaki, and H. Komatsu, “BnO at NTCIR-10 RITE: A Strong Shallow Approach and an Inference-based Textual Entailment Recognition System”, the 10th NTCIR Conference, 2013.
［5］ 池原 悟，宮崎 正弘，白井 諭，横尾 昭男，中岩 浩巳，小倉 健太郎，大山 史，林 良彦，「日本語語彙大系」，岩波書店，1997 年
［6］ 関口 宏司，「情報検索のための自然言語処理ツール群の開発［課題研究報告書］」，JAIST，2014 年
［7］ F. Bond, T. Baldwin, R. Fothergill, and K. Uchimoto, “Japanese semcor: A sense-tagged corpus of japanese”, In Proceedings of GWC-2012, 2012.
［8］ 隅田 飛鳥，吉永 直樹，鳥澤 健太郎，「Wikipedia の記事構造からの上位下位関係抽出」，自然言語処理，vol.16(3)，2008 年，P3-24,
［9］ 狩野 芳伸，「大学入試センター試験歴史科目の自動解答」，人工知能学会，2014 年

執筆者紹介 石　井　　愛（Ai Ishii）
2003 年日本ユニシス(株)入社．金融向けミドルウエアの開発・保守を担当．2009 年より R&D 部門にて，主に全文検索技術および自然言語処理の研究開発に従事.

宮　下　　洋（Hiroshi Miyashita）
1999 年日本ユニシス(株)入社．JP1 を使用した運用管理システムの構築・保守を担当．2013 年より R&D 部門にて，主に自然言語処理の研究開発に従事.