情報処理学会論文誌 Vol. 52 No. 12 3574–3587 (Dec. 2011)

Co-STAR：上位下位関係獲得のための共訓練アルゴリズム

呉 鍾勲[†1] 山田一郎[†1,†2] 鳥澤健太郎[†1]
デ・サーガ ステイン[†1] 橋本力[†1]

本論文では，構造化されたテキストと構造化されていないテキストを情報源として，単語間の上位下位関係を高精度に獲得する共訓練アルゴリズム Co-STAR（Co-training Style Algorithm for hyponymy Relation acquisition）を提案する．Co-STAR における 2 つの独立な上位下位関係の獲得処理は各々のテキストから抽出した異なる手がかりを利用し，得られた知識を交換することにより共訓練を行う．従来の共訓練とは異なり，Co-STAR は 2 つの異なる情報源の共通するインスタンスから効果的な学習データを獲得することで，精度の向上を実現する．実験では，構造化テキストとして日本語の Wikipedia を，非構造化テキストとして 5,000 万の Web ページを対象とし，大規模な上位下位関係獲得の処理を行い，Co-STAR の有効性を示した．また，Co-STAR はノイズの含まれる学習データを利用した場合でも頑健に動作することを確認した．

Co-STAR: A Co-training Style Algorithm for Hyponymy Relation Acquisition

Jong-Hoon Oh,[†1] Ichiro Yamada,[†1,†2]
Kentaro Torisawa,[†1] Stijn De Saeger[†1]
and Chikara Hashimoto[†1]

This paper proposes a co-training style algorithm called Co-STAR that acquires hyponymy relations simultaneously from structured and unstructured text. In Co-STAR, two independent processes for hyponymy relation acquisition – one handling structured text and the other handling unstructured text – collaborate by repeatedly exchanging the knowledge they acquired about hyponymy relations. Unlike conventional co-training, the two processes in Co-STAR are applied to different source texts and training data. We show the effectiveness of this algorithm through experiments on large-scale hyponymy-relation acquisition from Japanese Wikipedia and Web texts. We also show that Co-STAR is robust against noisy training data.

1. はじめに

単語間の意味的関係に関する知識は機械翻訳や情報検索などの自然言語を利用したアプリケーションにおいて重要な役割を果たす．しかし，複合語を含めた新しい単語も日々生まれており，また，特定の意味的関係を持つ単語の対も多数存在するため，多くの単語を網羅した知識を人手により構築・維持することは困難と考えられる．そこで，本論文では，単語の意味的関係の 1 つである上位下位関係[⋆1]を自動獲得する共訓練アルゴリズム Co-STAR（Co-training Style Algorithm for hyponymy Relation acquisition）を提案する．Co-STAR では，HTML テキストなどの構造化されたテキスト（構造化テキスト）と構造化されていないテキスト（非構造化テキスト）を対象として，2 つの独立な上位下位関係の獲得処理を行い，従来の共訓練のアプローチ[3] と同様に 2 つの分類器を利用して上位下位関係，より正確には上位下位関係を持つ単語対の集合を獲得する．

これまでにも，構造化テキストや非構造化テキストのどちらか 1 つを入力とする単語間の意味的関係獲得手法は，いくつか提案されている．それぞれ，以下に述べるように異なる手がかりを意味的関係獲得に利用している．

- 非構造化テキストから抽出可能な手がかり[1),5),7),11),17),21)]：構文パターン，分布類似度
- 構造化テキストから抽出可能な手がかり[10),14),16),19)]：Wikipedia のカテゴリ情報などの文書のトピック情報，文書階層，HTML タグ

さらに近年では，構造化テキストと非構造化テキストの両方から得られる手がかりを利用した手法が提案されている．Pennacchiotti ら[13)] は「アンサンブルセマンティックス」（Ensemble Semantics）という枠組みを提案した．Pennacchiotti らは構造化テキストと非

†1 情報通信研究機構
National Institute of Information and Communications Technology
†2 NHK 放送技術研究所
Science & Technical Research Laboratories, Japan Broadcasting Corporation

⋆1 本論文では，上位下位関係を，「A は B の一種です」，もしくは「A は B の一例です」のいずれかを満たす A と B の関係と定義する．前者の条件は，A と B がともに概念である場合で，たとえば「犬」と「哺乳類」がこの関係に該当する．後者は A がインスタンスで B が概念である場合で，たとえば「清水寺」と「お寺」がこの関係に該当する．

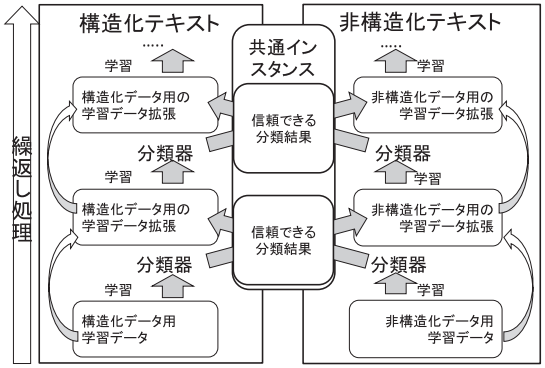


図 1　*Co-STAR* の概念図
Fig. 1　Concept of *Co-STAR*.

構造化テキストから得られる異なる手がかりすべてを素性として使って 1 つの分類器を学習し，学習した分類器を両テキストに適用することにより，意味的関係獲得の性能を改善した．Talukdar ら[20] は，構造化テキストと非構造化テキストをそれぞれ処理して得た意味的関係を持つ単語対をマージし，新たにランク付けすることによって，性能を改善した．

以下では，本論文で提案するアルゴリズム Co-STAR の特徴を以上に述べた先行研究との差に留意しつつ下記にまとめる．

Co-STAR は，半教師あり学習手法であり，構造化テキストと非構造化テキストに対する同期した 2 つの上位下位関係獲得処理で構成される（**図 1**）．各々の上位下位関係獲得処理は，簡単な手がかりによって上位下位関係を持つ候補の集合をそれぞれのテキストから抽出し，後述する学習方法によって得られた分類器によってそれらの上位下位関係候補を正例，すなわち「適切に上位下位関係候補を持つ単語対」と，負例，すなわち「上位下位関係候補を持たない単語対」とに分類する処理を行う．Co-STAR ではまず，人手で作成した初期学習データで構造化テキスト，非構造化テキストにそれぞれ適用される 2 つの分類器を学習し，ついで，各分類器による信頼できる分類結果，すなわち「正例と負例にそれぞれ一定以上の信頼度で分類された単語対」を相手側の新たな学習データとして利用し，学習と分類を繰り返す．各分類器は，構造化テキストと非構造化テキストのそれぞれから抽出可能な異なる手がかりを利用するため，同一の単語対に対してもまったく異なる分類結果を出力することがありうる．そうした差分をうまく利用する．つまり，片方の分類で十分な手がかりによって正確な分類結果が得られた単語対を，同一の単語対を高い信頼度をもって分類できない他方の分類器の学習データに加えることで，その学習データから抽出できる有力な手がかりが後者の分類器に新たに与えられ，その結果として精度の向上が期待できる．なお，上述したように単一の分類器を利用するアンサンブルセマンティックス[13] とは異なり，Co-STAR では 2 つの分類器が利用されることに注意されたい．アンサンブルセマンティックスの実装の詳細が明らかになっていないため，完全に公平な実験ではないが，Co-STAR では単一の分類器を構造化テキストと非構造化テキストの両方に適用した場合に比べて高い精度が得られることを後ほど実験により確認する．

なお，Co-STAR の開発においては，構造化テキストと非構造化テキストの組合せではなく，異なる 2 言語で書かれたテキストから上位下位関係を獲得する手法である言語横断共訓練[10] を参考とした．Co-STAR と言語横断共訓練の比較の詳細については 4 章を参照されたい．

上述したように，1 つの分類器の分類結果を他方の分類器の学習データに追加することで性能を向上させることが Co-STAR の狙いであるが，この学習データの追加を行うためには，そもそも追加されるべき単語対に関して両方の分類器で利用できる素性が得られることが必要条件となる．つまり，分類器が分類すべきデータを「上位下位関係候補」と呼ぶことにすると，両方の分類器に与えられる上位下位関係候補の共通部分に問題の単語対がない限り，片方の分類結果を他方の学習データに追加することはできない．以下では，このような上位下位関係候補の共通部分を「共通インスタンス」と呼ぶことにするが，見方を変えれば，Co-STAR においては，2 つの分類器がそうした共通インスタンスを介して知識のやりとりをしていることになる．ここで，この共通インスタンスの量，バリエーションが多ければ多いほど，やりとりされる知識の量が増え，より高い性能が実現されることが予想される．本研究においては，上位下位関係候補だけでなく，上位下位以外の関係候補（たとえば因果関係など）も利用することで，共通インスタンスの量を増大させ，性能の向上を実現している．

本研究では，上述の工夫とは別に，単語間の関係を記述したデータ（学習データ）を，自動作成ルールなどを利用して作成することにより，Co-STAR における初期の学習処理に要する人手による労力を激減させることが可能であることを示す．この場合，誤ったデータも学習データに含まれてしまうが，Co-STAR は，こうしたノイズに対しても頑健であることを示す．6 章における実験では，2 つの処理の一方の初期データに人手で与えた学習データ，他方に自動獲得したノイズの含まれる学習データを利用し，このような設定でも高精度に処理可能であることを確認した．

以下，2 章と 3 章で提案する Co-STAR の詳細を記し，4 章で関連研究との比較につい

てより詳細に説明する．5 章と 6 章で実験と評価を行い，7 章でまとめについて述べる．

2. Co-STAR

本章では，Co-STAR の詳細について述べるが，まず，2 つの分類器が知識のやりとりをする媒体となる共通インスタンスの構造について述べ，ついで，Co-STAR のアルゴリズムについて述べる．

2.1 共通インスタンス

構造化テキストと非構造化テキストの集合をそれぞれ S，U とする．本論文では構造化テキスト S として，日本語版 Wikipedia 記事を利用し，非構造化テキスト U として，Web の HTML タグを除外したテキスト，いわゆる平文を利用する．また，S と U から獲得した上位下位関係候補の集合を X_S，X_U とする．X_S は 3.1 節に詳しく述べるように Oh ら[10] に示されている手法を用いて Wikipedia の階層構造から抽出し，X_U は Ando ら[1] が提案した上位下位関係獲得のための語彙統語パターン（例：A などの B，A という B）により獲得する．

以下では共通インスタンスの集合を Y で表す．以下では**図 2** を参照しつつ，Y の構造について述べる．提案法では，2 種類の共通インスタンスを扱う．1 つ目は，S と U に共通して出現する上位下位関係候補集合とし，「関係候補共通インスタンス」（G）と呼ぶ（$G = X_S \cap X_U$）．2 つ目は，下記の条件を満たす単語対とし，「仮想共通インスタンス」（V）と呼ぶ．つまり，$Y = G \cup V$ となる．

- S または U の一方で上位下位関係候補である．
- 他方のデータで上位下位関係は持たないが何らかの意味的関係は持つ可能性が高い．

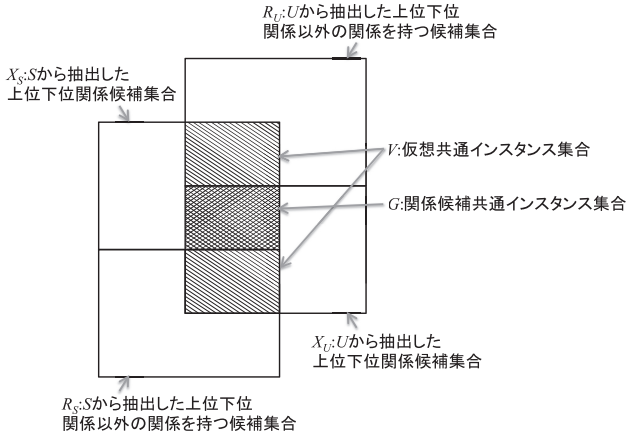


図 2　共通インスタンスの概要
Fig. 2　Common instances.

最初の条件は，V の要素が X_S または X_U に含まれることに相当する．2 つ目の条件を満たす S，U における集合，つまり，S で上位下位関係は持たないが何らかの意味的関係は持つ可能性が高い要素を R_S，U で上位下位関係を持たない可能性が高い要素を R_U と表す．R_S の要素は X_S，R_U の要素は X_U 中にそれぞれ含まれていないものであることに注意されたい．

ついで，より具体的に R_S はどういうものとなるのか説明する．Wikipedia のカテゴリシステムは Wikipedia の記事をトピックによって効率的に管理するためのものである．1 つの Wikipedia 記事は 1 つ以上のトピックを持ち，そのトピックはカテゴリ名で表現される．また，1 つのカテゴリには 1 つ以上の上位カテゴリが存在する．カテゴリ名になっている単語の対で，一方の単語が他方の親や祖父などの直系の先祖である場合，そうした単語対は上位下位関係になる可能性が高いことが知られている[18]．一方で，カテゴリ名になっている単語の対で片方が他方と直系の先祖関係を持たない場合，その単語対が上位下位関係になっていない可能性が非常に高い．R_S として本研究で設定した集合はこうしたカテゴリのシステムを使って生成する．**図 3** は Wikipedia カテゴリシステムから獲得した R_S の例を示す．すべてのカテゴリ名のペアは R_S の候補になり，そのうち，ある単語に対して，その親や祖父などの直系の先祖ではない（先祖関係ではない）ノードとの単語対が R_S の要素とする．たとえば，図 3 で，「感染症」と「タンパク質」は Wikipedia カテゴリシステムで先祖関係を持たないため，R_S に含まれるが，「感染症」と「新型インフルエンザ」は “「感染症」←「インフルエンザ」←「新型インフルエンザ」” あるいは “「感染症」←「ウイルス感染症」←「新型インフルエンザ」” という先祖関係になるため，R_S には含まれない．

R_U は，上位下位関係を表す語彙統語パターンでは抽出されず，かつ，上位下位関係以外の意味的関係を表すような語彙統語パターン（たとえば，因果関係を表す「A が B を起こす」，材料関係を表す「A でできた B」など）と共起する単語対のうち，X_U に含まれないものを利用する[⋆1]．最終的に $V = (X_S \cap R_U) \cup (X_U \cap R_S)$ となる（図 2）．

⋆1 本論文で非構造化テキストとして使われる 5,000 万 Web ページに現れるすべての語彙統語パターンのうち，上位下位関係を表す語彙統語パターンとして使われない約 118 万語彙統語パターンで取り出した約 1 億 3,000 万の単語対（単語対の異なり数は約 2,400 万）を R_U として使った．

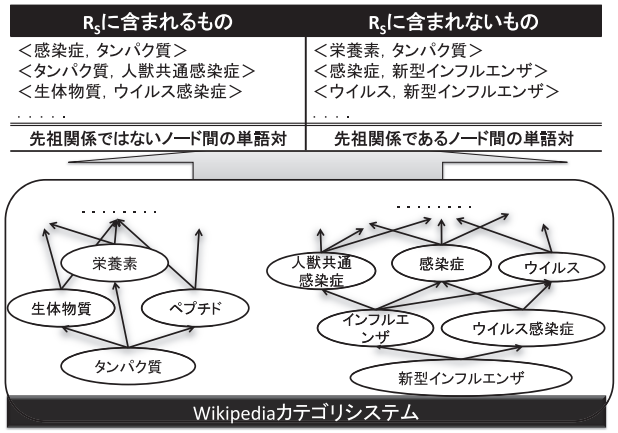


図 3　R_S の獲得手法とその例
Fig. 3　Examples of obtaining R_S instances.

V は，たとえ一方の文書に現れる手がかりに基づいて上位下位関係の候補となっていても，他方では上位下位関係になりにくい単語対の集合と考えられる．このような場合，単語対は上位下位関係ではないケースが多く，多くの V は分類器の学習データに負例として追加される．一方，G は両方の集合で上位下位関係候補となっているため，V と比較して上位下位関係になる可能性が高い．

共通インスタンス（$Y = G \cup V$）は，異なる 2 種類の上位下位関係獲得処理を協調させるための橋渡しとして利用される．この橋渡しを介して，構造化テキストと非構造化テキストに対する処理結果を相互に交換しながら処理を進めることにより，2 つの上位下位関係獲得処理が効果的に協調した共訓練を実現することができる．

2.2　アルゴリズム

本節では，Co-STAR のアルゴリズムについて具体的に述べる．まず，ノーテーションであるが，分類器 c が，上位下位関係候補からなる集合 X に含まれる各単語ペア $x \in X$ に対してクラスラベル $cl \in \{yes, no\}$（“yes” は上位下位関係を，“no” は上位下位関係ではないことを意味する），信頼度 $r \in R^+$ で割り当てるならば，それを $c(x) = \langle x, cl, r \rangle$ と表すことにする．本実験における分類器にはサポートベクタマシン（SVM）を利用し，超平面からの距離を信頼度 r の値とする．また，学習データ L から学習によって，分類器 c を得ることを $c = LEARN(L)$ で表す．

1: **Input**: 共通インスタンス (Y), 初期学習データ (L_S^0, L_U^0)
2: **Output**: 2 つの分類器 (c_S^n, c_U^n)
3: $i = 0$
4: **repeat**
5: $\quad c_S^i := LEARN(L_S^i)$
6: $\quad c_U^i := LEARN(L_U^i)$
7: $\quad CR_S^i := \{c_S^i(y) | y \in Y,\ y \notin L_S^i \cup L_U^i\}$
8: $\quad CR_U^i := \{c_U^i(y) | y \in Y,\ y \notin L_S^i \cup L_U^i\}$
9: $\quad L_U^{(i+1)} := L_U^i$
10: $\quad$ **for** each $\langle y, cl_S, r_S \rangle \in TopN(CR_S^i)$ and $\langle y, cl_U, r_U \rangle \in CR_U^i$ **do**
11: $\quad\quad$ **if** ($r_S > \alpha$ and $r_U < \beta$) or ($r_S > \alpha$ and $cl_S = cl_U$) **then**
12: $\quad\quad\quad L_U^{(i+1)} := L_U^{(i+1)} \cup \{(y, cl_S)\}$
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad L_S^{(i+1)} := L_S^i$
16: $\quad$ **for** each $\langle y, cl_U, r_U \rangle \in TopN(CR_U^i)$ and $\langle y, cl_S, r_S \rangle \in CR_S^i$ **do**
17: $\quad\quad$ **if** ($r_U > \alpha$ and $r_S < \beta$) or ($r_U > \alpha$ and $cl_S = cl_U$) **then**
18: $\quad\quad\quad L_S^{(i+1)} := L_S^{(i+1)} \cup \{(y, cl_U)\}$
19: $\quad\quad$ **end if**
20: $\quad$ **end for**
21: $\quad i = i + 1$
22: **until** stop condition is met

図 4　Co-STAR の疑似コード
Fig. 4　Co-STAR algorithm.

Co-STAR のアルゴリズムを**図 4** に示す．このアルゴリズムにおける繰返し処理では，

1）新しい学習データ（L_S^i, L_U^i）により新たな分類器（c_S^i, c_U^i）を学習し，各分類器による共通インスタンスの分類を行う（5〜8 行）．
2）ある条件を満たす共通インスタンスの分類結果を新たな学習データとして既存の学習データ（L_S^{i+1} と L_U^{i+1}）に追加する（9 行〜20 行）．

このアルゴリズムの初期段階では，2 つの分類器は人手によりラベルを付与された教師あ

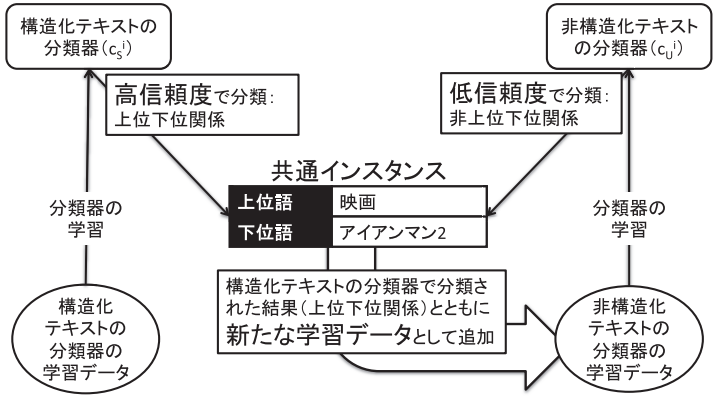
りデータ（L_S^0, L_U^0）によって，それぞれ学習を行う．4 行以降の繰返し処理では，まず，繰返し回数 i における分類器の学習を行い，作られた 2 つの分類器 c_S^i と c_U^i で共通インスタンスの分類を行う（5 行〜8 行）．この共通インスタンスの分類結果を $c_S^i(y)$ と $c_U^i(y)$ にする．CR_S^i と CR_U^i は i 回目の学習によって得られた分類器 c_S^i と c_U^i によって得られた共通インスタンスの分類結果のうち，c_S^i と c_U^i の学習で利用された学習データの和集合 $L_S^i \cup L_U^i$ に含まれない集合を示す．

9 行〜14 行は，学習データ L_U^i に新たな学習データを追加して，次回の繰返しにおける学習データ L_U^{i+1} を生成する手順を記述している．この処理では，分類器 c_S^i が分類器 c_U^i の教師役として動作する．より具体的に述べると，まず，$TopN(CR_S^i)$ は，r_S が CR_S^i における上位 N 位（5 章における実験では $N = 900$）に含まれる $c_S^i(y) = \langle y, cl_S, r_S \rangle$ の集合を示す．教師役として動作する分類器 c_S^i は，c_S^i の上位 N 位以内にある分類結果で，信頼度が一定値以上の値を持ち（$r_S > \alpha$），かつ，以下の条件を最低 1 つ満たす場合に，c_U^i に対して y のクラスラベルが cl_S であることを教示する．

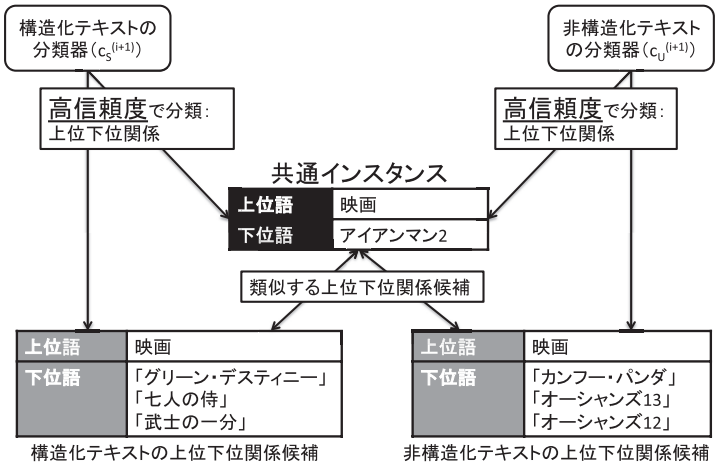
- c_U^i の分類結果の信頼度が一定値よりも小さい（$r_U < \beta$）
- c_S^i と c_U^i における y のクラスラベルが同じ（$cl_S = cl_U$）

1 つ目の条件によって，$c_S^i(y)$ と $c_U^i(y)$ のクラスラベルが異なる場合（$cl_S \neq cl_U$）でも，教師役となる分類器 c_S^i による判定結果を c_U^i に教えて，c_U^i の分類精度を向上させることができる．2 つ目の条件によって，c_S^i がある一定レベルの信頼度を持ち，かつ，$c_S^i(y)$ と $c_U^i(y)$ のクラスラベルが一致する場合，c_U^i の信頼度によらず教師役となる分類器 c_S^i による判定結果を c_U^i に教えることができる．さらに，2 つの条件によって，c_S^i と c_U^i がある一定レベルの信頼度で異なるクラスラベルを出力した場合（「$r_S > \alpha$ and $r_U > \beta$」，あるいは「$r_U > \alpha$ and $r_S > \beta$」）に，クラスラベルを教師に合わせて換えてしまうのを回避することができる．この場合，教師は何もせず，そのインスタンスを無視する．上記条件を満たす場合に，(y, c_S^i) を $(i+1)$ 番目の学習データ $L_U^{(i+1)}$ に追加する．15 行〜20 行は，9 行〜14 行の処理における S と U を入れ替えた処理について記述しており，c_U^i が c_S^i の教師役として動作する．

図 5 に非構造化テキストの分類器のために新たな学習データを追加する場合の例を示す．図 5 (a) に i 番目の学習データの追加処理を示す．図 5 (a) では，構造化テキストの分類器（c_S^i）は高信頼度で共通インスタンス「映画，アイアンマン 2」を正しく上位下位関係と分類し，一方，非構造化テキストの分類器（c_U^i）は同じ共通インスタンスに対して低信頼度で非上位下位関係と分類している．Co-STAR ではその 2 つの分類結果を比べて，c_S^i の分類結果が信じられると判定し，低信頼度で分類する c_U^i を改善するために「映画，アイアンマン 2」とその c_S^i の分類結果を c_U の既存学習データに新たな正例として追加する．その結果，図 5 (b) のように新たな学習データで学習された非構造化テキストの分類器（$c_U^{(i+1)}$）



(a) 非構造化テキストの方に新たな学習データを追加



(b) 追加された学習データの効果

図 5　非構造化テキストの方に新たな学習データを追加する例
Fig. 5　An example of selecting new training instances for c_U.

は共通インスタンス「映画，アイアンマン 2」を正しく上位下位関係と分類でき，その共通インスタンスと類似する非構造化テキストから抽出した上位下位関係に対しても正しく上位関係と分類できるようになる．この処理が構造化テキストの分類器においても行われ，両方の分類器を改善できる．また，繰返しによりその効果の範囲を広げることができる．

4 行〜22 行の繰返し処理は，2 つの分類器が共通インスタンスすべての分類に関して一定の合意が得られたと判定できる場合に終了する．この合意の度合いは，Wang ら[23] に用いられた式 (1) の $d(c_S^i, c_U^i)$ により判断する．

$$d(c_S^i, c_U^i) = |\sigma^i - \sigma^{(i-1)}|/|\sigma^{(i-1)}| \quad (1)$$

ここで σ^i は，i 番目の処理における，2 つの分類器が共通インスタンスすべてを分類した際の信頼度の差の平均を示す．$d(c_S^i, c_U^i)$ の値が 0.001 より小さくなった場合に，この処理を終了する．

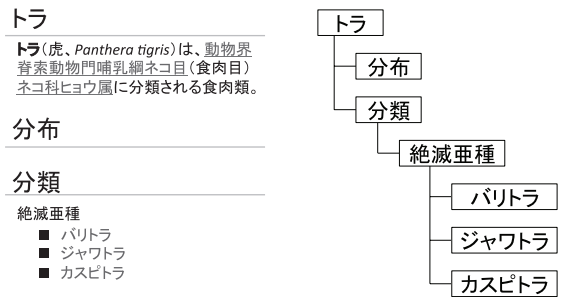
3. 上位下位関係獲得

本章では，Wikipedia および Web テキストから上位下位関係を獲得する 2 つの処理について説明する．各処理では，最初に上位下位関係の候補を抽出する．候補中には，多くの上位下位関係ではない単語対が存在する．そこで，この候補集合に対して，上位下位関係か否かを分類器で判定する．前述したように，本論文では，判定のために SVM[22] を利用する．以下では順を追って 2 つの処理の詳細を説明する．

3.1 Wikipedia からの上位下位関係獲得

日本語の Wikipedia を対象とした上位下位関係獲得は，Oh らの手法[10] に従う．Wikipedia の記事は記事タイトル，節タイトル，項目名などの階層構造を持ち，その階層構造は**図 6** (b) のように木構造に変換できる．木構造中の各ノードを上位語候補で，そのノードの下位にあるすべてのノードを下位語候補とすることにより，上位下位関係候補を抽出する．図 6 の例では，（トラ，分類），（トラ，バリトラ）などが，上位下位関係候補として抽出される．Wikipedia の全記事に対して，この処理を行うことにより，1,900 万ペアの上位下位関係候補が抽出された．

抽出した候補の多くは上位下位関係ではなく，一部に上位下位関係が含まれている程度である．たとえば，（トラ，分類）などは上位下位関係ではない．そこで抽出した全候補に対して，上位下位関係を表すか否かを SVM を使って判定する．この SVM 用の学習データを 2 つの分類器を繰り返し利用することで効率良く生成する手法が Co-STAR の新規性となる．この処理では，単語に含まれる形態素などの語彙的特徴，図 6 に示す木構造などの構造的特徴，そして Wikipedia の Infobox 情報などの特徴を SVM の素性として利用する．素性の詳細を**表 1** の「WikiFeature」に示す．



(a) 記事「トラ」の階層構造　(b) 図 6 (a) のに対応する木構造

図 6　Wikipedia 記事とその階層構造の例
Fig. 6　An example of layout structures of a Wikipedia article.

語彙的特徴は，上位下位関係にある上位語と下位語の語彙に関する特徴で，たとえば，図 6 で示した（トラ，バリトラ）などは，単語末尾の形態素「トラ」が一致している典型的な上位下位関係の特徴を持つ．単語に含まれる各形態素の表記や品詞を，語彙的特徴とする．

構造的特徴は，Wikipedia の木構造における上位語候補と下位語候補の出現位置の特徴を示す．構造上の距離は，たとえば図 6 の（トラ，分布）は直接の親子関係なので距離 1 という特徴が与えられる．「主な X」，「X のリスト」などの表現を含む節タイトルは，その構造の下位に X のインスタンスになる単語がよく現れ，X とそのインスタンスのペアが上位下位関係になることが多い．また，節タイトルなどに頻出する単語（「分類」，「歴史」など）は上位語としてはふさわしくない可能性が高い．構造的特徴では，ほかにも，階層構造の種類や，木構造における配置，上位語候補と下位語候補の親ノード，子ノードの表記などを特徴として利用する．

Wikipedia の Infobox には，記事タイトルに対する上位語と考える Infobox 名[⋆1] と，その属性名，属性値が記述されていることが知られている[2),24)]．この情報から，上位下位関係候

⋆1 たとえば，Wikipedia 記事「クリスティアーノ・ロナウド」では，Infobox 名「サッカー選手」を持つ Infobox が存在している．この Infobox 名は「クリスティアーノ・ロナウド」の上位語として考える．

表 1　素性（WikiFeature, WebFeature）：*hyper*，*hypo* は，それぞれ上位下位関係候補の上位語候補，下位語候補を示す．

Table 1　Feature sets (WikiFeature and WebFeature): *hyper* and *hypo* represent hypernym and hyponym parts of hyponymy relation candidates, respectively.

対象	種類	詳細
Wikipedia	語彙的特徴	*hyper*, *hypo* の表記;
		hyper, *hypo* に含まれる形態素の表記と品詞
（“WikiFeature”）	構造的特徴	*hyper*, *hypo* の構造上の距離;
		節タイトルなどに頻出する語彙統語パターンとの一致の有無（「主な X」,「X のリスト」など);
		節タイトルなどに頻出する単語か否か（「参照」,「分類」など);
		hyper, *hypo* の階層構造の種類（「節タイトル」,「項目名」など);
		hyper, *hypo* の木構造における配置（「ルートノード」,「リーフノード」など);
		hyper, *hypo* の親ノードと子ノードの表記
	Infobox	Wikipedia infobox における属性情報
Web テキスト	語彙的特徴	*hyper*, *hypo* の表記;
		hyper, *hypo* に含まれる形態素の表記と品詞
（“WebFeature”）	パターン	*hyper*, *hypo* を抽出した語彙統語パターン;
		語彙統語パターンと *hyper*, *hypo* ペア間の PMI
	共起	*hyper* と *hypo* 間の PMI
	単語のクラス	*hyper*, *hypo* の属する単語クラス

補の上位語候補（*hyper*），もしくは下位語候補（*hypo*）が，Infobox から獲得した（Infobox 名，属性名，属性値）の属性値に該当する場合，その属性値に対する「Infobox 名，属性名」を素性として与えて，各候補の意味的情報として利用する．たとえば，Wikipedia 記事「クリスティアーノ・ロナウド」の Infobox「サッカー選手」には，属性名「所属チーム名」と属性値「レアル・マドリード」が記述されている．「レアル・マドリード」が *hyper*，もしくは *hypo* として出現した場合，「レアル・マドリード」に「サッカー選手，所属チーム名」という素性を付与できる．なお，Infobox の素性は，上位下位関係候補の hyper，もしくは hypo が Infobox から獲得した（Infobox 名，属性名，属性値）の属性値に該当する場合のみに付与されることに注意されたい．

3.2　Web テキストからの上位下位関係獲得

日本語の Web テキストから上位下位関係を獲得する処理では，1 億 Web ページから構成される TSUBAKI コーパス[15] のうち，5,000 万ページを対象とする．このコーパスは

表 2　上位下位関係候補抽出に使った語彙統語パターン：*hyper*，*hypo* は，それぞれ上位下位関係候補の上位語候補，下位語候補を示す

Table 2　Lexico-syntactic patterns for hyponymy relations.

hypo など（の）*hyper*
hypo に似た *hyper*
hypo のような *hyper*
hypo という（と言う）*hyper*
hypo と呼ばれる *hyper*
hypo 以外の *hyper*

JUMAN [⋆1] により形態素解析が行われている．まず，Ando ら[1] が提案した**表 2** の「*hypo* などの *hyper*」,「*hypo* という *hyper*」といった上位下位関係を表す典型的な語彙統語パターンを利用して上位下位関係候補を獲得する．TSUBAKI コーパスに，上位下位関係を表す語彙統語パターンを適用することにより，600 万ペアの上位下位関係候補が抽出された．この候補にも，上位下位関係ではないものが大量に含まれるため，抽出した候補が上位下位関係を表すか否かを SVM により判定する．この処理で使用する SVM の素性を表 1 の「WebFeature」に示す．WebFeature では WikiFeature と同様に，上位下位関係のための語彙的な手がかりを認識するために語彙的特徴を使う．

また，使用する語彙統語パターンによって，獲得できる上位下位関係候補のカバレージや，その候補が正しい上位下位関係となる割合が異なる[1] ため，この語彙統語パターンも素性として利用する．また，ある上位下位関係候補が，複数の語彙統語パターンによって獲得されている場合，その候補は，上位下位関係である可能性が高いと考えられる．このような特徴をとらえるため，パターンも素性として利用する．さらに，語彙統語パターンから抽出した候補は，候補中の単語の一般の Web 文書での頻度に対して，1 文中での共起頻度が大きいほど適切な上位下位関係になる傾向があることが予備実験より分かった．そこで，上位語候補，下位語候補が同一文で共起する割合を指標とした Pantel ら[12] が用いた PMI（Point-wise Mutual Information）を素性として利用する．

単語の意味クラスも，単語間の意味的関係獲得において有益な情報と考えられている[5),9)]．何らかの方法で得られた名詞クラスが与えられたとして，語彙統語パターンによって抽出された単語対が，そうした一群の名詞クラスの中の同じ名詞クラスに属する場合や，上位語の名詞クラスと下位語の名詞クラスのペアが頻繁に共起するような場合は，上位下位関係にな

⋆1 http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html

る傾向が予備実験で確認された．そのため，上位語と下位語候補の単語クラスを素性として与える．単語クラスは，処理対象とする TSUBAKI コーパスから獲得した 4 億個の単語間係り受け関係（名詞が助詞を介して動詞を修飾する 3 つ組など）を情報源として，Kazama らの手法[9] により，50 万個の単語（複数の形態素で構成される複合名詞も含む）を 500 個のクラスに分類する処理を行い，この結果を利用した．Kazama らの手法を利用したクラスタリングでは，単語 n が属する単語クラス nc の確率 $p(nc|n)$ が与えられる．本論文では，De Saeger ら[5] が用いた条件である「$p(nc|n) > 0.2$ を満たす単語クラス nc を，単語 n の属するクラスとする」を利用する．

4. 関 連 研 究

本研究がベースとした従来手法として，半教師あり学習手法である共訓練[3]，言語横断共訓練[10] が存在する．**表 3** に共訓練，言語横断共訓練，そして，Co-STAR の違いを示す．共訓練では同一の学習データから素性生成を行い，そこから異なる素性セットを人手で選び，2 つの分類器をそれぞれの素性セットで学習する．そうした素性セットの差分をうまく利用することで，全体的な性能の向上を図っている．なお，共訓練では 2 つの分類器が同じデータを対象にするため，すべての関係候補インスタンスが共通インスタンスとして使われる．言語横断共訓練は異なる言語によって書かれたまったく異なるデータを使い，2 言語のデータをそれぞれ対象とする 2 つの分類器を各言語に固有の素性で学習する．言語横断共訓練の共通インスタンスは翻訳できる 2 言語の関係候補インスタンスのみになる．ここでもやはり，言語の差，素性の差が性能向上につながっている．また，これら 3 種の方法はそれぞれ，データの分割/非分割の仕方がまったく異なり，同じ設定では利用できないことに注意されたい．もちろん，これらをすべて組み合わせて，さらなる高性能の実現を目指すことは論理的には可能である．

表 3　共訓練，言語横断共訓練，Co-STAR の相違点
Table 3　Differences among co-training, bilingual co-training, and Co-STAR.

	共訓練[3]	言語横断共訓練[10]
処理対象	同一データ	言語によって異なるデータ
素性の分割	人が素性を分割	対象言語により分割
共通インスタンス	関係候補共通インスタンス	関係候補共通インスタンス（翻訳による）
	Co-STAR（提案手法）	
処理対象	構造によって異なるデータ	
素性の分割	対象データにより分割	
共通インスタンス	関係候補共通インスタンスと仮想共通インスタンス	

Co-STAR は言語横断共訓練と異なり，処理対象と分類器の学習に使われる素性が言語ではなくテキストの種類（構造化テキスト，または非構造化テキスト）によって分割されており，また，テキストの種類の差にともなって素性セットも異なることになる．また，共通インスタンスとして，Co-STAR のみ仮想共通インスタンスを考慮する点が，他のアプローチと大きく異なる．

また，提案手法とはまったく異なるアプローチとして，単語間の関係獲得手法の一種であるアンサンブルセマンティックス[13] がある．これは，構造化テキストと非構造化テキストの両方を最終的に単一の分類器，ないしはランカによって候補の分類やランク付けを行っている．一方で Co-STAR では，2 種のデータを同時に分類できる単一の分類器は存在しない．なお，次章で述べる実験においては，このアンサンブルセマンティスクに対応する手法を実際に実装し，Co-STAR と比較する実験を行っている．

なお，大規模の Web テキストから単語間の意味的関係を獲得する手法として NELL[4]，Probase[25]，TextRunner[6] がある．NELL は繰返し学習（iterative learning）の枠組みを持ち，あるテキスト集合から機械学習に基づく複数の上位下位関係獲得プロセスで単語間の上位下位関係を獲得し，獲得された単語間の上位下位関係を知識として統合する．Probase は NELL と同様に繰返し学習の枠組みを持つ．Hearst[7] が提案した上位下位関係のための語彙統合パターンに基づいて上位下位関係を獲得し，その結果から上位下位関係の上位語となる確率，あるいは下位語となる確率などの確率的な知識を推定する．この知識を，繰返し学習における次の上位下位関係の獲得に適用することにより上位下位関係獲得の性能を改善した．名詞句間の意味的関係を獲得する手法である TextRunner では，名詞句間の意味的関係を 2 つの名詞句とその間の文字列によって表現し，自己教師あり学習（Self-supervised learning）によって学習された分類器により「信頼できる結果」と「信頼できない結果」のいずれかと判定する．最後に「信頼できる結果」と判定された意味的関係とその意味的関係が獲得された文章に基づいた統計値を用いて意味的関係のランク付けを行う．NELL と Probase は，学習プロセスの適用を繰り返すことで上位下位関係獲得の精度を改善した点で Co-STAR と類似しているが，構造化テキストと非構造化テキストという 2 種類のテキストを利用している本手法は非構造化テキストのみを利用しているそれらの手法とは異なる．

5. 実　　験

実験のために，まず，構造化テキストと非構造化テキストに対して，各分類器用の学習データと，Co-STAR のパラメータを最適化するための開発用データ，そして手法の評価を行うためのテストデータを作成する．

構造化テキストに対しては，2009 年 7 月バージョンの日本語 Wikipedia を利用する．Wikipedia から抽出した約 1,900 万の上位下位関係候補のうち，24,000 ペアをランダムに選択して人手により上位下位関係であるかを判断した．このうち 20,000 ペアを初期分類器のための学習データとして利用し，残りの 4,000 ペアを開発用データとテストデータに等分割した．つまり，互いに共通部分を持たない 2,000 ペアをそれぞれ開発用データ，テストデータとした．これらを「Wiki セット」と呼ぶ（**表 4** の Wiki セット参照）．非構造化テキストに対しては，TSUBAKI コーパス[15] の約 5,000 万日本語 Web ページを利用した．この非構造化データから抽出した約 600 万の上位下位関係候補から，9,500 ペアをランダムに選択し，人手により上位下位関係であるかを判断した．このうち 7,500 ペアを，初期分類器のための学習データとして利用し，残りを等分割し，1,000 ペアづつをそれぞれ開発用データとテストデータとした．これらを「Web セット」と呼ぶ（表 4 の Web セット参照）．

表 5 に実験に使われた「構造化テキストからの上位下位関係候補」，「非構造化テキストからの上位下位関係候補」，「関係候補共通インスタンス」，「仮想共通インスタンス」の量を示す．

表 4　Wiki セットと Web セットのデータ量
Table 4　Statistics on test set.

種類	Wiki セット	Web セット
学習データ	20,000	7,500
開発用データ	2,000	1,000
テストデータ	2,000	1,000

表 5　実験に用いられた上位下位関係の候補と共通インスタンスの数
Table 5　Statistics on hyponymy relation instances and common instances.

構造化テキストからの上位下位関係候補（X_S）	約 1,900 万
非構造化テキストからの上位下位関係候補（X_U）	約 600 万
関係候補共通インスタンス（G）	約 67,000
仮想共通インスタンス（V）	約 576,000

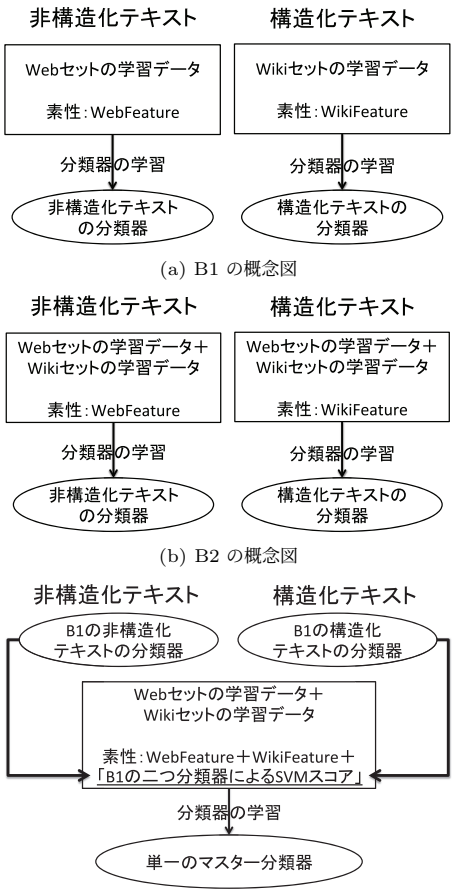
Co-STAR の各分類器は，TinySVM [⋆1] の 2 次の多項式カーネルを使用した．開発用データを利用した予備実験によって精度が最良となったときの $\alpha = 1$（高信頼度と判断する下限のしきい値 α），$\beta = 0.3$（低信頼度の上限のしきい値 β），$TopN$=900（各繰返し処理で新たに学習データに追加する数の上限の $TopN$）を Co-STAR のパラメータとして評価実験に使用した．評価では，上位下位関係と判定されたペアが正しい割合を示す適合率（P），テストデータ中の全上位下位関係に対して正しく抽出できた割合を示す再現率（R），そして，適合率と再現率の調和平均である F 値（F）を利用する．

5.1 結　　果

実験では後述する 6 つの手法を比較する．B1〜B3 は繰返し処理を行わない手法であり，学習の際の素性の違い（表 1 における WikiFeature と WebFeature）と，学習データの違いによる影響を検討するためのものである．**図 7** はそのような違いを示している．B1 と B2 では，2 つの別々の分類器を学習するが，B3 では，すべての素性と学習データを統合し，1 つのマスタの分類器を学習する．これらの 3 つの分類器は，人手によって用意された Wiki セットと Web セットの学習データによって学習する．B3 は，アンサンブルセマンティクス[13] の近似と考えることができる．以下に，B1〜B3 で使用する学習データと素性をまとめる．

- B1：2 つの独立した分類器で構成（図 7 (a) を参照）．Wiki セットの学習データと WikiFeature により構造化テキスト S の分類器を学習し，Web セットの学習データと WebFeature により非構造化テキスト U の分類器を学習する．
- B2：2 つの独立した分類器で構成（図 7 (b) を参照）．Wiki セットの学習データと Web セットの学習データを統合して，合計 27,500 個の学習データを作成する．この学習データにより，WikiFeature を素性とした S の分類器と，WebFeature を素性とした U の分類器を学習する[⋆2]．
- B3：B1 の 2 つの独立した分類器に，マスタ分類器となる 1 つの分類器を追加した構成（図 7 (c) を参照）．マスタ分類器のための素性として，表 1 に示した全素性（WikiFeature と WebFeature）のうち抽出可能なもの，そして，B1 の 2 つの分類器の SVM ス

⋆1 http://chasen.org/˜taku/software
⋆2 Web セット（または Wiki セット）の学習データは，もし，単語対が Wikipedia（または Web）にも出現していれば，WikiFeature（または WebFeature）を抽出できる．Wikipedia（または Web）に出現していない場合は，共通した素性（語彙的特徴）のみを使用する．



(a) B1 の概念図

(b) B2 の概念図

(c) B3 の概念図

図 7　B1，B2，B3 の概念図
Fig. 7　System architecture of B1, B2, B3.

コア⋆1（アンサンブルセマンティクス[13] の候補抽出スコアを近似するために用いる．）

⋆1 B1 の 2 つの分類器の学習データに対する SVM スコアは，学習データにおける 10-fold のクロスバリデーションによって計算された．

表 6　各手法による評価結果の比較
Table 6　Comparison of different systems.

	Web セット			Wiki セット		
	P	R	F	P	R	F
B1	84.3	65.2	73.5	87.8	74.7	80.7
B2	83.4	69.6	75.9	87.4	79.5	83.2
B3	82.2	72.0	76.8	86.1	77.7	81.7
BICO	N/A	N/A	N/A	84.5	**81.8**	83.1
Co-B	**86.2**	63.5	73.2	**89.7**	74.1	81.2
Co-B∗	85.5	69.9	77.0	89.6	76.5	82.5
Co-STAR	85.9	76.0	80.6	88.0	**81.8**	**84.8**
Co-STAR∗	83.3	**80.7**	**82.0**	87.6	**81.8**	84.6

を使う．マスタ分類器の学習データとして，B2 と同様に統合した 27,500 個の学習データを使う．マスタ分類器の出力を，最終的な分類結果とする．

これら 3 つの手法に加え，繰返し学習の処理を行う言語横断共訓練[10]（BICO）と提案手法である Co-STAR，そして，Co-STAR アルゴリズムにおいて仮想共通インスタンスを使用しない手法（Co-B）の比較実験を行う．Co-B と Co-STAR 間の比較は，異なる構成の共通インスタンスを使った影響を調べるためである．さらに Co-B と Co-STAR では，B1 と B2 を初期分類器とした 2 種の実験を行った．以下に各手法の詳細を記す．

- BICO：2 つの言語（日本語と英語）の Wikipedia を対象として，各言語における処理が協調して上位下位関係を獲得する言語横断共訓練アルゴリズム[10] を実装．20,000 ペアの英語の学習データと，20,000 ペアの日本語の学習データ（Wiki セットと同じ）を，人手に与えて，2 つの処理における分類器を学習する．BICO は日英 Wikipedia を処理対象としているため，非構造化テキストを対象とした実験を行うことができない．
- Co-B：Co-STAR アルゴリズムにおける仮想共通インスタンスの効果を明確にするために，共通インスタンスとして関係候補共通インスタンス（67,000 インスタンス）のみを使用したもの⋆2 ．
- Co-STAR：提案手法．関係候補共通インスタンスと仮想共通インスタンスの両方を使用（643,000 インスタンス）．

表 6 に 6 つの手法による実験の評価結果を示す．表 6 の Co-B と Co-STAR は B1 を初

⋆2 Co-B は，2 つの分類器が同一データを解析しながら協調するという点では，オリジナルの共訓練[3] と見なすことができる．

期分類器とした結果を，Co-B∗ と Co-STAR∗ は B2 を初期分類器とした結果を示している．

表 6 では，B2 と B3 の結果が B1 より F 値で良いことが分かる．B2 と B3 は各分類器の学習データとして 27,500 ペアをそれぞれ使用しており，S の分類器の学習データとして 20,000 ペア，U の分類器の学習データとして 7,500 ペアを使用した B1 と比較して，多くの学習データを使用している．この学習データ量の違いが精度向上に貢献していると考えられる．B2 と B3 は，異なる数の分類器で構成され，それぞれ異なる素性で学習しているにもかかわらず，F 値において，ほぼ同じ評価結果であった．

Co-STAR の評価結果は，アンサンブルセマンティクスに類似する B3 より優れている．さらに，Co-STAR は BICO より少量の学習データ（Co-STAR は総計 27,500 ペア，BICO は総計 40,000 ペア）を利用しているにもかかわらず，Co-STAR は BICO より良い精度が得られている．Co-B と Co-STAR 間における精度の差分は，自動生成した仮想共通インスタンスを利用する効果を示している．初期分類器の種類（B1 または B2）にかかわらず，Co-STAR の F 値は Co-B を上回っており，仮想共通インスタンスを関係候補共通インスタンスとともに利用することは，Co-STAR における 2 つの分類器間の協調を効果的にしていることが分かる．

提案手法は，他の手法と比較して F 値で 1.4%〜8.5%優れていた．最終的に，適合率が 90%以上となるように開発用データによって SVM のしきい値を調整した結果 90%を表すしきい値で TSUBAKI コーパスからは約 43 万ペア，Wikipedia の全記事からは約 462 万ペアの上位下位関係を獲得することができた[⋆1]．

5.2 考　　察

Web から獲得した上位下位関係と Wikipedia から獲得した上位下位関係は異なる特徴を持っている．Web テキストからの上位下位関係獲得プロセスは，上位下位候補に関わる共起情報，語彙統合パターンと上位下位候補間の共起情報，そして単語クラスタリングの結果に基づいた単語クラス情報などの統計的な情報を用いて上位語候補候補の分類を行うため，Web テキストに高頻度で現れる単語対で構成される上位下位関係を獲得する傾向がある．一方，Wikipedia からの上位下位関係獲得プロセスは，Wikipedia が持つ階層構造情報，Infobox 情報などなどの構造から取り出せる情報を用いて上位下位関係候補の分類を行うため，対象となる語の頻度に対する依存度は Web からの場合に比べて低い．つまり，Wikipedia からは Web には低頻度でしか出現しない語に関する上位下位関係が抽出される傾向がある．実際に，Wikipedia から獲得された約 462 万ペアの上位下位関係が持つ下位語のうち，約 73%が本論文の実験に使われた Web テキストに対して 5 以下の出現頻度を持ち，約 67%が Web テキストに出現しない単語であった．つまり，Web テキストから獲得できる上位下位関係と Wikipedia テキストから獲得できる上位下位関係の種類は大きく異なり，両テキストから獲得した上位下位関係を統合することにより，より多様な上位下位関係が得られるといえる．

6. 自動作成した学習データを利用した実験

Co-STAR では，構造化テキスト用と非構造化テキスト用の分類器のために，2 つの学習データを人手により用意しなければならない．他の教師あり学習における問題と同様に，学習データの構築には相当な労力を要する．そこで，一方の初期データに人手で与えた学習データ，他方に自動獲得した学習データを利用し，このような設定でも高精度に処理可能であることを確認した．本実験では Wikipedia の記事における定義文[⋆2]とカテゴリ名を利用して自動獲得し，非構造化テキスト（Web テキスト）のための学習データとして利用した．

この手法では，まず，Wikipedia の定義文特有の上位下位関係を明示する語彙統語パターン「A は B である」，「A は B の一種です」などを利用することにより，Wikipedia の記事タイトルに対する上位語候補を獲得する[8),19)]．次に，定義文から獲得した上位語候補と，Wikipedia の記事に付与されたカテゴリ名とを比較する．このカテゴリ名と定義文から獲得した上位語候補の最終形態素が一致する場合，これらを記事タイトルの上位語と考え，上位下位関係を作る．最終的に，その上位下位関係が Web テキストから獲得した上位下位関係候補に含まれている場合，Web テキストのための学習データの正例として使う．**図 8** は Wikipedia 記事「新型インフルエンザ」から正例「ウイルス感染症，新型インフルエンザ」を自動作成する処理例を示している．図 8 では，記事「新型インフルエンザ」の定義文に「A は B である」という語彙統語パターンを適用し，記事タイトル「新型インフルエンザ」の上位語候補「インフルエンザ感染症」を獲得する．次に，この上位語候補と「新型インフルエンザ」のカテゴリ名の最終形態素を比較する．カテゴリ名「ウイルス感染症」と定義文から獲得した上位語候補「インフルエンザ感染症」の最終形態素が一致するため，「新型インフルエンザ」の上位語として「ウイルス感染症」と「インフルエンザ感染症」が抽出され

⋆1 TSUBAKI コバースの場合は 0.23，Wikipedia の場合は 0.1 を SVM スコアにおけるしきい値として使った．その結果，TSUBAKI コバースからの上位下位関係候補の 7%が，Wikipedia からの上位下位関係候補の約 24%が上位下位関係と判定された．

⋆2 Wikipedia の記事における第 1 文を定義文とする．

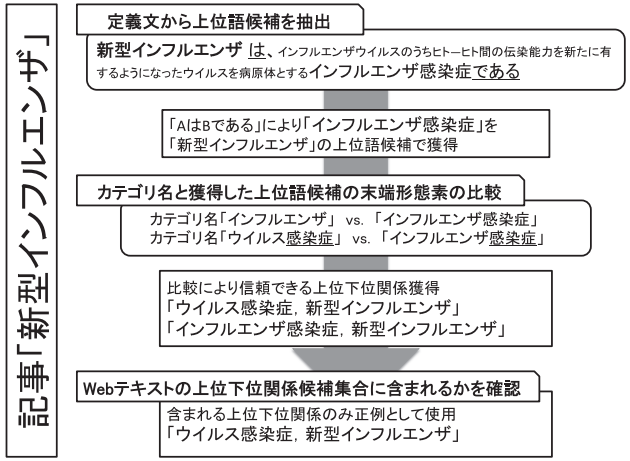


図 8　正例の自動作成手法
Fig. 8　A way to automatically generate positive instances.

る．最終的にこの上位下位関係が Web テキストから抽出した上位下位関係候補集合（2 章の「図 2 の X_U」に相当）にあるか否かを調べ，集合にあるもののみを正例として使う（約 15,000 インスタンス）．

負例は，Web テキストから獲得した「上位下位関係候補集合」（「図 2 の X_U」に相当）と Wikipedia から獲得した「上位下位関係を持たない可能性が高い単語対の集合」（「図 2 の R_S」に相当）の積集合に相当する仮想共通インスタンス（「図 2 の $X_U \cap R_S$」に相当し，約 293,000 インスタンスを含んでいる）から選択した．構造化テキスト（Wikipedia）の学習データは，手作業で作成したものを利用し，自動獲得は行わない．

自動獲得した学習データにはノイズが含まれる．また，学習データのサイズは手作業で生成したデータに比べて大きい．2 つの学習データを手作業で生成した前章における実験との比較のため，この自動生成した学習データから 7,500 ペア（手作業で生成した学習データと同数）を任意に選択した．ここで，正例と負例の割合は 1 : 4 とした[⋆1]．**表 7** に使用した学習データの一部を示す．

⋆1 開発用データを用いて正例と負例の割合を 1 : 2，1 : 4，1 : 5 とした予備実験を行い，その F 値が最良であった 1 : 4 を用いた．

表 7　自動作成した学習データの例：*が付いている例はノイズを表す
Table 7　Examples of automatically generated training data.

正例		負例	
上位語	下位語	上位語	下位語
うどん	釜揚げうどん	ウイルス	カビ
アクション映画	スピード	企業	電話機
動物	野生動物	ミュージシャン	オーストラリア
化学反応	加水分解	外国語*	ハングル*
ウイルス感染症	新型インフルエンザ	湖沼*	琵琶湖*
アニメ	∀ガンダム	医療機関	肝炎
タンパク質	シトクロム	島	カリブ海
水*	氷*	基本	学校教育
法*	乾布摩擦*	黄色	熱帯魚
ホラー映画	オーメン	首相	人形
俳優	中村俊介	残留農薬	ワイン

表 8　自動獲得した学習データによる結果
Table 8　Results with automatically generated training data.

	Web セット			Wiki セット		
	P	R	F	P	R	F
B1	81.0	47.6	60.0	**87.8**	74.7	80.7
B2	80.0	55.4	65.5	87.1	79.5	83.1
B3	82.0	33.7	47.8	87.1	75.6	81.0
Co-STAR	**82.2**	60.8	69.9	87.3	80.7	83.8
Co-STAR∗	79.2	**69.6**	**74.1**	87.0	**81.8**	**84.4**

Web テキストに対しては自動獲得した学習データを，Wikipedia に対しては手作業で与えた学習データを用い，B1〜B3，Co-STAR の実験を行った．評価結果を**表 8** に示す．表 8 から，Co-STAR は自動獲得した学習データを使用しても B1〜B3 より F 値が高く，ノイズに対しても頑健に処理できることが分かる．

7. ま と め

本論文では，構造化テキストと非構造化テキストに対する同期した 2 つの上位下位関係獲得処理で構成される共訓練アルゴリズム Co-STAR を提案した．各処理では，構造化テキストと非構造化テキストのそれぞれから抽出可能な異なる手がかりを利用するため，同一の単語対に対してまったく異なる分類結果を出力することがありうる．Co-STAR はこのような差分を利用し，1 つの分類器の分類結果を他方の分類器の学習データに追加することで性

能を向上させる．実験では，単一の分類器を構造化テキストと非構造化テキストの両方に適用した場合と比べて提案手法がより高い精度で上位下位関係を獲得できることを確認した．

また，「関係候補共通インスタンス」という上位下位関係候補集合だけでなく，「仮想共通インスタンス」という上位下位以外の関係候補も利用することによって，Co-STAR の性能向上ができることを示した．

さらに，単語間の関係を記述したデータ（学習データ）を，自動作成ルールなどを利用して作成することにより，Co-STAR における初期の学習処理に要する人手による労力を激減させることが可能であることを示した．この場合，誤ったデータも学習データに含まれてしまうが，Co-STAR は，こうしたノイズに対しても頑健であることを示した．

参 考 文 献

1) Ando, M., Sekine, S. and Ishiza, S.: Automatic Extraction of Hyponyms from Japanese Newspaper Using Lexico-syntactic Patterns, *LREC'04: Proc. 4th International Conference on Language Resources and Evaluation* (2004).
2) Auer, S. and Lehmann, J.: What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content, *ESWC'07: Proc. 4th European Semantic Web Conference*, Springer, pp.503–517 (2007).
3) Blum, A. and Mitchell, T.: Combining labeled and unlabeled data with co-training, *COLT'98: Proc. 11th Annual Conference on Computational Learning Theory*, pp.92–100 (1998).
4) Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, Jr., E.R. and Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning, *AAAI'10: Proc. 24th Conference on Artificial Intelligence* (2010).
5) DeSaeger, S., Torisawa, K., Kazama, J., Kuroda, K. and Murata, M.: Large Scale Relation Acquisition Using Class Dependent Patterns, *ICDM'09: Proc. IEEE International Conference on Data Mining Series*, pp.764–769 (2009).
6) Etzioni, O., Banko, M., Soderland, S. and Weld, D.S.: Open information extraction from the web, *Comm. ACM*, Vol.51, pp.68–74 (2008).
7) Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora, *COLING'92: Proc. 14th Conference on Computational Linguistics*, pp.539–545 (1992).
8) Kazama, J. and Torisawa, K.: Exploiting Wikipedia as External Knowledge for Named Entity Recognition, *EMNLP-CoNLL'07: Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp.698–707 (2007).
9) Kazama, J. and Torisawa, K.: Inducing Gazetteers for Named Entity Recognition by Large-Scale Clustering of Dependency Relations, *ACL-08: HLT: Proc. 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.407–415 (2008).
10) Oh, J.-H., Uchimoto, K. and Torisawa, K.: Bilingual Co-Training for Monolingual Hyponymy-Relation Acquisition, *ACL-09: IJCNLP: Proc. Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp.432–440 (2009).
11) Pantel, P., Crestan, E., Borkovsky, A., Popescu, A.-M. and Vyas, V.: Web-Scale Distributional Similarity and Entity Set Expansion, *EMNLP'09: Proc. 2009 Conference on Empirical Methods on Natural Language Processing*, pp.938–947 (2009).
12) Pantel, P. and Ravichandran, D.: Automatically Labeling Semantic Classes, *HLT-NAACL'04: Proc. Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp.321–328 (2004).
13) Pennacchiotti, M. and Pantel, P.: Entity Extraction via Ensemble Semantics, *EMNLP'09: Proc. 2009 Conference on Empirical Methods in Natural Language Processing*, pp.238–247 (2009).
14) Ravi, S. and Pasca, M.: Using structured text for large-scale attribute extraction, *CIKM'08: Proc. 17th ACM Conference on Information and Knowledge Management*, pp.1183–1192 (2008).
15) Shinzato, K., Shibata, T., Kawahara, D., Hashimoto, C. and Kurohashi, S.: Tsubaki: An open search engine infrastructure for developing new information access, *IJCNLP'08: Proc. 3rd International Joint Conference on Natural Language Processing*, pp.189–196 (2008).
16) Shinzato, K. and Torisawa, K.: Extracting hyponyms of prespecified hypernyms from itemizations and headings in web documents, *COLING'04: Proc. 20th International Conference on Computational Linguistics*, pp.938–944 (2004).
17) Snow, R., Jurafsky, D. and Ng, A.Y.: Semantic taxonomy induction from heterogenous evidence, *COLING-ACL'06: Proc. 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pp.801–808 (2006).
18) Suchanek, F.M., Kasneci, G. and Weikum, G.: Yago: A Core of Semantic Knowledge, *WWW'07: Proc. 16th International World Wide Web Conference*, pp.697–706 (2007).
19) Sumida, A. and Torisawa, K.: Hacking Wikipedia for Hyponymy Relation Acquisition, *IJCNLP'08: Proc. 3rd International Joint Conference on Natural Language Processing*, pp.883–888 (2008).
20) Talukdar, P.P., Reisinger, J., Pasca, M., Ravichandran, D., Bhagat, R. and Pereira, F.: Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random

Walks, *EMNLP'08: Proc. 2008 Conference on Empirical Methods on Natural Language Processing*, pp.582–590 (2008).
21) Van Durme, B. and Pasca, M.: Finding Cars, Goddesses and Enzymes: Parametrizable Acquisition of Labeled Instances for Open-Domain Information Extraction, *AAAI'08: Proc. 23rd Annual Conference on Artificial Intelligence*, pp.1243–1248 (2008).
22) Vapnik, V.N.: *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA (1995).
23) Wang, W. and Zhou, Z.-H.: Analyzing Co-training Style Algorithms, *ECML'07: Proc. 18th European Conference on Machine Learning*, pp.454–465 (2007).
24) Wu, F. and Weld, D.S.: Autonomously semantifying Wikipedia, *CIKM'07: Proc. 16th ACM Conference on Information and Knowledge Management*, pp.41–50 (2007).
25) Wu, W., Li, H., Wang, H. and Zhu, K.Q.: Towards a Probabilistic Taxonomy of Many Concepts, *VLDB'11: Proc. 37th International Conference on Very Large Data Bases* (2011).

(平成 23 年 4 月 11 日受付)
(平成 23 年 9 月 12 日採録)

呉　　鍾勲

独立行政法人情報通信研究機構ユニバーサルコミュニケーション研究所情報分析研究室専攻研究員．1998 年韓国成均館大学校情報工学科卒業．2005 年 KAIST（韓国科学技術院）電子電算学科電算学専攻博士課程修了．同年 KAIST 研究員を経て，情報通信研究機構に専攻研究員として着任．博士（工学）．自然言語処理の研究に従事．

山田　一郎（正会員）

1991 年名古屋大学工学部情報工学科卒業．1993 年同大学大学院修士課程修了．博士（情報科学）．同年 NHK 入局．1996 年より NHK 放送技術研究所にて自然言語処理を利用した情報抽出，メタデータ生成の研究に従事．2003〜2004 年スタンフォード大客員研究員．2008〜2011 年情報通信研究機構専門研究員．現在，NHK 放送技術研究所主任研究員．映像情報メディア学会，言語処理学会各会員．

鳥澤健太郎（正会員）

1992 年東京大学理学部卒業．1994 年同大学大学院修士課程修了．1995 年同大学院博士課程中退．同年同大学院助手．1998 年科学技術振興事業団さきがけ研究 21 研究員兼任（2002 年まで）．北陸先端科学技術大学院大学助教授を経て，2008 年より独立行政法人情報通信研究機構言語基盤グループ，グループリーダー．2011 年より同機構情報分析研究室室長，現在に至る．博士（理学）．自然言語処理の研究に従事．日本学術振興会賞等受賞．言語処理学会，人工知能学会，ACL 各会員．

デ・サーガステイン

2006 年北陸先端科学技術大学院大学知識科学研究科博士課程修了．北陸先端科学技術大学院大学研究員を経て，2007 年に情報通信研究機構に入所．2008 年に NICT MASTAR プロジェクト言語基盤グループに専攻研究員として着任．自然言語処理を用いた知識獲得の研究に従事．

橋本　　力（正会員）

1999 年福島大学教育学部卒業．2001 年北陸先端科学技術大学院大学博士前期課程修了．2005 年神戸松蔭女子学院大学大学院博士後期課程修了．京都大学大学院情報学研究科産学官連携研究員を経て，2007 年山形大学大学院理工学研究科助教，2009 年より独立行政法人情報通信研究機構専攻研究員．2011 年京都大学大学院情報学研究科博士後期課程修了．現在に至る．自然言語処理の研究に従事．博士（言語科学，情報学）．言語処理学会，ACL 各会員．