情報科学論文からの意味関係抽出に向けたタグ付けスキーマ

建石 由佳† 仕田原 容‡ 宮尾 祐介† 相澤 彰子†
†国立情報学研究所 , ‡フリーランス
†{yucca, yusuke, aizawa}@nii.ac.jp, ‡yo.shidahara@gmail.com

1 はじめに

学術論文の検索ではキーワード検索以上の機能が求められるようになっている．出版年, 著者, 分野, など書誌情報からの検索のほか, Google Scholar[1]やMicrosoft Academic Search[2]などでは引用関係から関連研究を探すこともできる．さらに, 研究動向の解析 ([2][8] など) や, 他の論文をどういった文脈で引用しているのか ([1] など）という情報を抽出する研究も行われている．

我々は, さらに論文の意味内容に踏み込んだ検索を実現するために, 論文中に出てくるモノとモノの意味関係を同定し, それにより例えばあるシステムについて「そのしくみの記述か」「その評価か」「それを他に応用したのか」などが区別できることを目指す．そのために, 文中で記述されたモノとモノの関係を構造化してタグ付けしたコーパスを作成した．従来, 文単位で方法, 結果, 評価などを区別するコーパスが作られている（[3] など）が, 本研究では文内部の構造にタグ付けを行い, 情報科学分野の論文アブストラクトのすべての記述内容について意味的構造を与えるタグ付けスキーマを構築した．

研究動向の解析では文内の語句の意味役割を手掛かりとする．例えば, [2] では, Technology, Effect, Attribute, Value に対応する Named Entity を同定したコーパス [5] を利用している．また, [8] では Focus, Technique, Domain の抽出を行っている．しかし, 実際は情報科学分野における概念は単純な名詞句では表現されないことが多い．例えば, 「ビデオ撮影による覗き見攻撃に対しても安全性を確保可能にする認証手法」では, [8] でいう Domain にあたるものは「ビデオ撮影による覗き見攻撃に対しても安全性を確保可能にする (こと)」となる．[8] では Dependency Parsing を行うことにより Domain などにあたるものの内部構造を獲得しているが, 本研究ではこの内部についても意味的な関係を構造化することを試みた．

従来研究と本研究との最大の違いは, 論文テキストの情報を網羅的に構造化する点にあり, 従来の情報抽出やテキストマイニングを超えてテキストの意味内容の解析に踏み込んでいるといえる．

[1]http://scholar.google.co.jp/
[2]http://academic.research.microsoft.com/

2 スキーマ

2.1 方針

情報科学分野に特化して扱いやすい構造にしたかったため, Framenet[6] など言語学に基づく枠組みを用いることはしなかった．一方で, この分野には生命科学分野における Gene Ontology[4] のように広く使われる標準オントロジーがない．また, 生命科学のように「すでにあるものの性質を研究する」のではなく,「ある機能を果たすためのしくみを作る」研究が多いという情報科学分野の特徴により, 固定したオントロジーが作りにくい．したがって, タグ付けを行いながら並行して必要な関係そのものを定義していった．

情報科学論文は, トピックとなるシステムや技法を記述した, 記述対象の体系の中の世界と, その体系の外から記述する筆者の世界の二重の構造を持っていると考える．あるシステム, 手法などの記述について,「しくみの記述」「応用」は体系の中の世界の出来事であり,「評価」は体系の外から行われているといえる．研究動向の解析のために抽出されてきた情報は, 体系の中の世界を対象とするものであり, 我々はそれに加えて, その評価という, 体系の中と外にまたがる情報をとりあげた．また, 記述されている研究の背景となる応用分野や関連研究などにも関係づけを行った．この結果, 論文中の語と語の関係をほぼもれなく構造化することになった．

2.2 タグセット

現行タグセットは 16 種類の関係と, 3 種類のエンティティ（関係の「項」）を持つ．

2.2.1 エンティティ

エンティティを示す語句に対しては, TERM, OBJECT, MEASURE のいずれかのタグを付与する．TERM は下記の OBJECT, MEASURE 以外で項となるもので, 名詞とは限らない．OBJECT は, システム名など実体を持つものの名前である．専門分野に限らず, 国名, 人名なども OBJECT とする．ただし, ア

ルゴリズム, スキーマ, 言語の名前は OBJECT とせず, 中に人名などを含んでいても全体で TERM とする. MEASURE は評価, 価値判断を示す語（「良い」, 「高性能」など）, 数値・数量表現のほか, 可能性（「できる」, 「不可能」など）, 必要性（「したい」, 「必要」）を表す語に対して付与される. エンティティは入れ子にせず, 複合語の場合は最も長い範囲を取ることとする. ただし, 可能性・必要性を表す語は前後から分離する.

2.2.2 エンティティ間の関係

関係はエンティティ間の有向関係で, すべて二項関係あるいはその組み合わせとしてあらわす. 現行タグセットでは

システムの振る舞いに関するもの PERFORM (動作主体), APPLY_TO (アルゴリズム, 手段, 意図した結果, 目的), RESULT (意図しない結果, 副作用, 因果関係), INPUT (入力, 材料), OUTPUT (出力, 成果物), TARGET (動作対象で, それ自身が変化しないもの), ORIGIN (起点), DESTINATION (着点)

性質に関するもの CONDITION（実験条件, 状況), ATTRIBUTE (属性, 特徴), STATE (（筆者以外の）他のものに関する評価・感情)

評価に関するもの EVALUATE (（筆者による）評価結果), COMPARE (評価の際の比較対象)

その他 SUBCONCEPT (上位-下位, 全体-部分, クラス-インスタンス), EQUIVALENCE (定義, 略称, 照応), SPLIT (カッコなどで分断された語句の前後をつなぐ)

の 16 種類の関係を定義する.

ここで, プログラムの機能などを表す動詞句内部でも, 上の関係を用いて動詞の格要素と動詞との間の関係を構造化する. これは, ちょうど動詞句で記述されるイベントをそれを実現するための「機械」ととらえ, それに対する入力, 出力, 手段, 副作用などがあると考えることに相当する. 単純な例では, 「作成したテストデータ」では, 成果物にあたる「テストデータ」を OUTPUT とし, 「バイオメトリクスとゼロ知識証明を組み合わせたプロトコル」では, 組み合せのパーツである「バイオメトリクス」, 「プロトコル」を INPUT, 組合せの成果物である「プロトコル」を OUTPUT とする. また, 「ビジネス環境の変化」のようなケースでは, 「変化を起こす機械」に古いビジネス環境を入れたら新しいビジネス環境が出力された, と考え, 「ビジネス環境」と「変化」との間に INPUT,OUTPUT 関係を二重につける. 一方, 「観察する」のような, 対象の変化を伴わない動詞の場合, 対象は TARGET

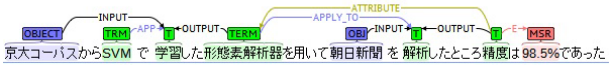

図 1: タグ付け例

で関係づける. また, 「車を加速する」のように, 一時的（可逆的）な属性の変化が起こる場合も, 対象は TARGET で関係づける. DESTINATION は動作の着点のほか, 付加する際にもとからあったもの（例 :「IP パケットに認証情報を付加」の「IP パケット」）や表示先（例 :「手のひらに映像を表示」の「手のひら」）を含む. ORIGIN は DESTINATION と対になるもので, 削除の際の削除元（例 :「ネットワークから不特定者を排除」の「ネットワーク」）を含む.

CONDITION は時間, 実験条件など「体系の中にあるが, 記述されるシステムなどの外部にある制約」に用い, ATTRIBUTE は記述されるシステムなどそのものの属性や特徴に用いる. STATE はシステムのユーザ, ゲームのプレーヤなど, 「記述対象の体系の中の人」がシステムやゲームに対して持っている評価, 感情に使う. ユーザなどのふるまいでも, 主体的な動作については STATE でなく PERFORM 関係をつける. プログラムの動作の場合, その動作がユーザの行いたいことと同じ, 即ち, プログラムがユーザが何かを行うための手段となっている場合は APPLY_TO, そう考えるには無理がある場合は PERFORM を用いる. 実際は多くのケースが APPLY_TO となる.

EVALUATE と COMPARE は筆者 (論文で記述される体系の外の人）の視点から見た評価にかかわるものである. EVALUATE は評価対象, COMPARE は評価の際の比較対象である. 筆者の主体的動作, すなわち, （システムなどについて）「述べる」「提案する」などについては無視する.

関係の方向は, APPLY_TO は手段などから適用対象, INPUT などは入出力などからシステム（格要素から動詞）, ATTRIBUTE, CONDITION は性質/条件からそれを持つもの, SUBCONCEPT は下位 (部分）から上位（全体）に向けて付けると定める. EVALUATE は評価対象, COMPARE は比較対象から評価表現に向けて付ける. SPLIT と EQUIVALENCE はテキスト上で後ろから前に付ける.

表 1: エンティティのタグ付け結果

	数	%
一致	1639	86.3
ラベルのみ違う	48	2.5
範囲の最初のみ違う	15	0.8
範囲の最後のみ違う	44	2.3
部分的に重なる	0	0
片方がもう片方を真に含む	1	0.0
重ならない	152	8.1
計	1899	100.0

表 2: 関係のタグ付け結果

	数	%
一致	1108	47.1
ラベルのみ違う	190	8.1
方向のみ違う	41	1.7
ラベルと方向が違う	65	2.8
その他	947	40.3
合計	2351	100.0

3 コーパス作成実験

情報処理学会論文誌アブストラクト 41 件を 2 名でタグ付けしながらスキーマを作成した．さらに, スキーマの評価を兼ねて別の 30 件（6601 形態素[3]）に同じ 2 人で独立にタグ付けし, 結果を比較した．コーパス作成は brat[7] を用いて行った．

作業者間の一致 (単純一致) は表 1,2 に示すように, エンティティで 86.3 %, 関係で 47.1 %となった．ラベル間の Confusion Matrix は, エンティティで表 3, 関係で表 4 のようになった．両表で, X は片方の作業者が関係をつけなかったものと, エンティティの範囲（表 4 では関係するエンティティの一方以上の範囲）の最初と最後が両方ずれたのものの合計であり, それ以外ものはそれぞれのラベルに含まれる．また, 表 4 においては方向は無視している．

表 3: エンティティラベル間の Confusion Matrix

	TERM	OBJECT	MEASURE	X
TERM	1479	11	29	54
OBJECT	2	13		
MEASURE	15		197	20
X	60		20	

エンティティ間のラベルの揺れでは次のようなことがわかった : 1) TERM と OBJECT の間では, ある名前が具体的なシステムの名前なのか, スキーマなどの名前なのかわからないと正確にタグがつけられないため判断が揺れる．2) TERM と MEASURE の間では,「問題」などの語や,「可能性」のような評価表現（この場合は「可能である」）を名詞化したものに関して判断が揺れる．1) については, brat の Web 検索機能などを通して「どんなものの名前」なのかを確認することが求められる．2) については,「問題になる」,「問題である」のように, 筆者の価値判断が読み取れる場合は MEASURE とし,「問題を解決する」のように,「問題」という語自体に価値判断が含まれていない場合は TERM とすることにした．この結果「○○が問題である．この問題を解決するために…」のような文章では, 2 つの「問題」に別種のタグがつくことになる．また,「可能性を持つ」では「可能性」といっただけでは評価にならない,「持つ」という表現が出てきて初めて評価となっていると考え,「可能性」を TERM,「持つ」を MEASURE として EVALUATE 関係で結ぶことにした．

表 4 からは, INPUT-OUTPUT-TARGET の 3 者と, ATTRIBUTE-CONDITION, ATTRIBUTE-SUBCONCEPT, ATTRIBUTE-EVALUATE, CONDITION-EVLUATION 間で揺れが多くみられることがわかる．

INPUT-OUTPUT-TARGET については, INPUT-OUTPUT を二重付けするか片方だけにするかに揺れや,「一貫性の確保」のような「ある状態が維持される」例で, TARGET の絡む揺れが多くみられた．これについては, 動詞ごとに格要素のタグ付けの目安を作ることで対処できるが,「変化の有無」「変化の可逆性」の判断が難しいことが原因にあると考えられ, INPUT-OUTPUT の二重付けをするケースと TARGET の統合, すなわち,「入力とも出力とも取れない, または, 両方であると取れる」ものを TARGET としてしまうことも含め, 詳細な検討が必要である．

ATTRIBUTE-CONDITION では「一時的に利用」「現状の脅威」など, 時間に関するものと,「特定のアプリケーション」「各種の数値」など「種別」に関するものに分かれた．統一的に前者を CONDITION, 後者を ATTRIBUTE にすることとした．ATTRIBUTE-SUBCONCEPT については「色の配列」「他の端末」のような種別に関する例と,「センサの電源」のようにシステムのコンポーネントともその機能とも取れる例に分かれた．前者については統一的に ATTRIBUTE とし, 後者については部分なのか機能なのかで区別するができないときは SUBCONCEPT 優先ということにした．ATTRIBUTE-EVALUATION については,「評価であるのか, 単に性質を述べているのか」(例 :「柔軟なアクセス制御」など) 判断が微妙なケースが多くみられた．これについては「筆者の目指すもの」と「実際にできた評価」を分け, 前者を ATTRIBUTE, 後者を EVALUATION とするなど別の視点が必要な可能性もあるが, 新たな揺れを生む可能性もあり, 検討課題である．

CONDITION-EVLUATION については, 1 例を除き「複数の」「1 回の」などの数量にかかわる表現であった．これについては数量は統一的に評価表現扱い (EVALUATE の対象）とすることにした．

一致性を重視するならば, INPUT-OUTPUT-TARGET (動作の対象), APPLY_TO-PERFORM (動作の主体), CONDITION-ATTRIBUTE-EVALUATION-SUBCONCEPT-STATE (何かの性質) をそれぞれ統合することも考えらえるが, 今後の検討課題とする．

タグ付けされたテキストを観察すると, 不一致となる関係として, 評価に絡むものが目立った．特に, 何かの「属性」を評価している,「通信量を抑えたルーティング方式」のような表現やイベント内でのモノの役割を評価している,「実現が難しかった実験」,「オーバレイネットワーク構築手法が検証に有益」などの表現で, どこからどこへ EVALUATE 関係をつければよい

[3]デフォルトの MeCab0.994 による

表 4: 関係ラベル間の Confusion Matrix

	APP	ATT	CMP	CND	DST	EQU	EVL	IN	ORI	OUT	PER	RES	SPL	STA	SUB	TAR	X
APPLY_TO	130	2					1	2		1		2					52
ATTRIBUTE	13	183		5	1		11	1		4		1			2	3	68
COMPARE			12														2
CONDITION	2	11		53			13										30
DESTINATION		1			38				1		1				1		8
EQUIVALENCE	1	1				71									3		25
EVALUATE		4				1	174			2		1			1	1	40
INPUT	5	2		2	1		11	137		21					3	20	36
ORIGIN									2						1		1
OUTPUT	3	4		1		1	2	13		183					1	4	92
PERFORM	6	1									15					3	4
RESULT	3			1								29					37
SPLIT													1				2
STATE	1	2															2
SUBCONCEPT	2	11		5		5	1			1		1			85		39
TARGET	5			5	1		2	4		4						36	14
X	90	63	3	36	16	21	70	29	2	55	10	37			55	8	

のかに揺れがあった.

これらの場合は, 意味的に直接評価されているものを EVALUATE の対象とし, 他の語からは評価表現を無視した場合に付けられる関係を, 評価されているものを示す語につけることとした. 例えば, 「通信量を抑えたルーティング方式」では, 筆者の意図としては方式を評価しているのであるが, 実際に抑えられたという評価を受けているのはその方式の通信量であるから, 「通信量」から「抑えた」へ EVALUATE 関係をつけ, 通信量は方式の属性にあたるから「通信量」から「ルーティング方式」へ ATTRIBUTE 関係をつける. 同様に, 「実現が難しかった実験」(「実験の実現」が難しい) は「実現」から「難しい」へ EVALUATE, 「実験」から「実現」へ OUTPUT, 「オーバレイネットワーク構築手法が検証に有益」(検証の手段としてオーバレイネットワーク構築処方が有益) では「オーバレイネットワーク構築手法」から「有益」へ EVALUATE, 「オーバレイネットワーク構築手法」から「検証」へ APPLY_TO の関係をつけることとした.

4 おわりに

現在, 評価に使用した 30 件の不一致個所を修正したものとタグ付けガイドラインを限定的に公開している[4]. また, 今回の実験の分析に基づいて修正したスキーマを用いて再度コーパス作成実験を行う予定である.

本研究は, 論文の検索という応用を想定して, 言語学理論に基づかない枠組みとして出発したが, INPUT, OUTPUT などに基づく我々のスキーマは, 格要素で表わされるものの移動と変化に基づいて意味を記述する語彙概念構造 [9] に類似したものとなっている. 今後, これらの意味表現間の関係について精査し, 相互変換のしくみなどを作りたい.

[4]連絡先：yusuke@nii.ac.jp

参考文献

[1] Athar A. and Teufel S. Context-enhanced citation sentiment detection. In *Proceedings of NAACL-12*, 2012.

[2] Fukuda S. et al. Extraction and visualization of technical trend information from research papers and patents. In *Proceedings of the 1st International Workshop on Mining Scientific Publications*, 2012.

[3] Liakata M. et al. Corpora for conceptualisation and zoning of scientific papers. In *Proceedings of LREC-10*, 2010.

[4] Michael A et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, Vol. 25, No. 1, pp. 25–29, 2000.

[5] Nanba H. et al. Overview of the patent mining task at the NTCIR-8 workshop. In *Proceedins of the 8th NTCIR Workshop Meeting*, 2010.

[6] Ruppenhofer J et al. Framenet ii: Extended theory and practice. *Berkeley FrameNet Release*, Vol. 1, , 2006.

[7] Stenetorp P. et al. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL*, 2012.

[8] Gupta S. and Manning C. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th IJCNLP*, 2011.

[9] 松林優一郎他. 語彙概念構造による意味役割の形式化と複数役割の割り当て. 言語処理学会第 17 回年次大会, 2011.