複層意味フレーム分析 (の簡略版) を使った意味役割タグづけの現状
タグづけデータから派生する言語資源の紹介を中心に

黒田 航　李 在鎬　渋谷 良方　井佐原 均
独立行政法人 情報通信研究機構 知識創成コミュニケーション研究センター

1　はじめに

NICT は 2005 年から複層意味フレーム分析 (以下 MSFA と略す) [5] を使った意味タグづけのプロジェクトを行なっている．この論文では意味タグづけの結果から自動生成される言語資源とそれを利用した言語処理を紹介する．意味処理を実現するたの資源を提供するという自明な目標の他に，§3.2 で述べるように，これまでの言語処理で想定されている表示の枠を改訂するという間接的な狙いもあった．

1.1　派生的言語資源の概要

MSFA と後述の MSFA Lite [6] で文書集合 D にフレーム基盤の意味タグを行なうと次の A-C のデータベースが派生的に構築される:

(1) a. フレームとそれが生起する文の集合との対の一覧 (データ A)
b. フレーム要素名 (=状況単位の意味役割) とそれが生起する位置情報の対の一覧 (データ B)
c. 単一のフレーム要素に対応する (不) 連続な部分文字列の集合 (データ C)

B は D 中の部分文字列の集合からフレーム要素名の集合への写像を元にし，C はフレーム要素名の集合から D 中の部分文字列の集合への写像を元にしている．B は条件つきだが概念単位での情報検索を可能にし，C はフレーズ規模のシソーラス/表現辞書に相当する新しいタイプの言語資源である．

1.1.1　公開サイト

(2) の三つのサイトのそれぞれで，異なる文書集合に対する (場合によって異なった版の) MSFA から自動生成された A, B, C のデータが公開されている ((2b) のデータは (2a) のデータを含有している):

(2) a. `http://www.kotonoba.net/~focal/cgi-bin/hiki/hiki.cgi?FrontPage`
b. `http://www.kotonoba.net/~focal/cgi-bin/hiki1/hiki.cgi?FrontPage`
c. `http://www.kotonoba.net/~focal/cgi-bin/hiki2/hiki.cgi?FrontPage`

参考までに (2b) には 2008/01/29 の時点で，215 文のタグづけ結果が登録され，それから 1000 個程度のもっとも粒度の粗いフレーム，6000 個程度のフレーム要素，4000 個程度の語句がデータベース化されている (有意味な部分文字列のデータベース化は不完全であり，実現されているのはエンコードされている情報の一部)．

タグづけの対象の利用の自由度に基づいて分類すると，(2b) と (2c) の対象は著作権で保護されている京大コーパスの文を含むものであり，今のところ完全に自由な利用はできない (それには第一著者が発行するアカウントが必要) が，(2a) の対象は著作権で保護されていない日英対訳文対応付けデータ (EJTAD) [8] であり，自由に利用可能である．

タグづけ仕様の区別から見ると，(2c) のデータがもっとも古い版の MSFA を，(2b) が §1.1.2 で説明する新しい版の MSFA Lite を使用している (現在もっとも盛んに開発されているのは (2b) である)[1]．(2a) は MSFA Full と MSFA Lite のデータが混在している．(2a) の MSFA Lite データは (2b) にも重複して登録されているので，Lite 版の全データは (2b) に登録されている．

対象言語の区別から見ると，(2c) は日本語文へのタグづけデータのみだが，(2a) と (2b) はまだ少数ながら EJTAD の英語文へのタグづけデータも含んでいる．

1.1.2　MSFA Lite の仕様

意味タグづけは 2005 年以来，基本的には [5] が定義する MSFA の範囲内で行なって来たが，主にタグづけ速度を上げる目的で 2007 年半ばから大きく仕様を変更した MSFA Lite を導入し，それを使った作業に移行した．

MSFA Lite は MSFA の原典版 (MSFA Full) の仕様を次の 4 つの点で変更したものである:

(3) タグづけ対象の選別: MSFA Full では状況 σ を (フレーム f として) 喚起する要素は，品詞の別を問わず，すべて f.EVOKER として記述していたが，MSFA Lite では喚起要素を (i) 動詞と形容 (動) 詞の一部と (ii) 動詞派生の名詞 (e.g., “**選択**肢”, “**受験**者”), (iii) 動詞派生の形容動詞の一部 (e.g., **破壊**的な)

(4) タグづけの複雑さの低減: MSFA Full では記述に使われるフレーム数に上限はなかった．MSFA Lite では ((3) に加えて) 述語一つについて，原則として一つのフレームだけを認定する[2]．

(5) フレーム f の支配要素 f.GOV(ORNOR) を語彙的に実現する述語の明示化の義務化: MSFA Full では個々のフレームの記述は f.GOV の語彙的異なりを捨象した抽象的なレベルで行われていた．MSFA Lite では f.GOV を代表的に実現している述語 p を (NULL を含めて) 少なくとも一つ特定することを義務づけた (f.GOV を語彙的に実現す

[1] (2c) の開発は事実上休止している．

[2] ただし，この原則は頻繁に必要に応じて放棄されるだけでなく，次の体系的な例外も存在する: 述語の意味が「字義通り」でない用法では 字義通りの意味 (= source) と実際に理解される意味 (= target) の両方を常に明示するように取り決めをしておかないと，どのフレームを認定するべきかでタグづけ作業者が混乱することが頻繁に起こる．一般に**タグづけ作業者に要求される語義の択一選択は単に「酷な課題」であるだけでなく，適切な制約を設けないと (自由度が高すぎて) 実行不可能な課題である可能性がある**(これは特に動詞の意味記述の場合に真である)．実際，MSFA Full では語義の択一選択が原因で生じるタグづけ結果の「不一致」の影響を軽減するためにフレームの複数指定が可能であるよう仕様が決められており，かつ奨励されている．

る述語は通常はフレーム名に表れている動詞と同一だが，それが一致しない場合も少なくない)[3]．

(6) MARKER の扱いの変更: MSFA Full では *e*.MARKER (*e* はフレームの要素) の値の指定は意図的に行なっていないが，MSFA Lite では (5) を前提にできるので，*e*.MARKER の値を (通常は (複合)(格) 助詞として) 明示化している[4]．

(7) の MSFA Lite の実例を図 1 に示す:

(7) ロシア南部チェチェン共和国の首都グロズヌイに進攻したロシア軍は三十一日、首都中心部を装甲車などで攻撃、大統領官邸など数カ所が炎上した。[S-ID:950101004-002]

MSFA の際，係り受け解析の結果は (意図的に) 参照していない．形態素の区切りはフレーム要素の境界に合わせてあり，独自のものである (JUMAN の境界と一致しない場合，それを細分化を示す記号 “~” や誤った分離の結合を表わす記号 “+” で明示してある)．この理由は §3.2 で後述する．

1.2 派生的言語処理: 文意の分解を媒介にした換言

一般に文 *s* の MSFA (Lite) を使った意味タグづけは，*s* の (非線型性 [4] を保存したまま) 意味フレームという形式で記述される基本的言明へ分解する処理と見なせる．MSFA Full ではそのための情報が不足しているが，MSFA Lite では (5)–(6) によってその情報が提供されている．従って，MSFA Lite の意味タグづけの結果である基本的言明の集合のおのおのを適当に補完，整形することで，*s* を単 (重) 文の集合 $\{t_1, t_2, \ldots, t_n\}$ に分解する処理が実現できる (これは換言処理の一種と言える)．この処理の試験的な実装を行なった．結果は §3.1 で述べる．

2 タグづけから派生する言語資源の詳細

2.1 FrameList と FrameElementList

文書への状況=フレーム単位の意味役割 =フレーム要素 (frame element: fe) タグづけにより，タグづけに使われた状況のデータベース化，状況を構成する意味役割のデータベース化が自動に行われる (以後，しばしばフレームを *f* で，フレーム要素を *e* で表示)．これらは FrameList と FrameElementList というインターフェイスで管理されている．

フレーム要素はタグづけされた文の部分文字列の位置と対になっているので，タグづけされた文書集合 *D* を検索対象とした意味検索が可能となる[5]．図 1 の MSFA を例に取り上げると，“首都グロズヌイ” に ⟨先 [進出の]⟩ や ⟨対象 [攻撃の]⟩ という fe が，“チェチェン共和国” に ⟨対象 [侵略の]⟩ という fe が付与されている．fe の実現値は FrameElementList というインターフェイスから検索できる．例えば ⟨対象 [侵略の]⟩ と ⟨対象 [攻撃の]⟩ が ⟨対象⟩ という見出しから，⟨先 [進出の]⟩ が ⟨先⟩ という見出しから検索できる．

2.2 Word/PhraseList: “語句のシソーラス” の雛形

フレーム要素 $e_1, e_2, \ldots, e_n$ でタグづけしたデータは自動処理され，単一の e_i に対応する部分文字列がすべてデータベース化されている．これは Word/PhraseList というインターフェイスで管理されている．

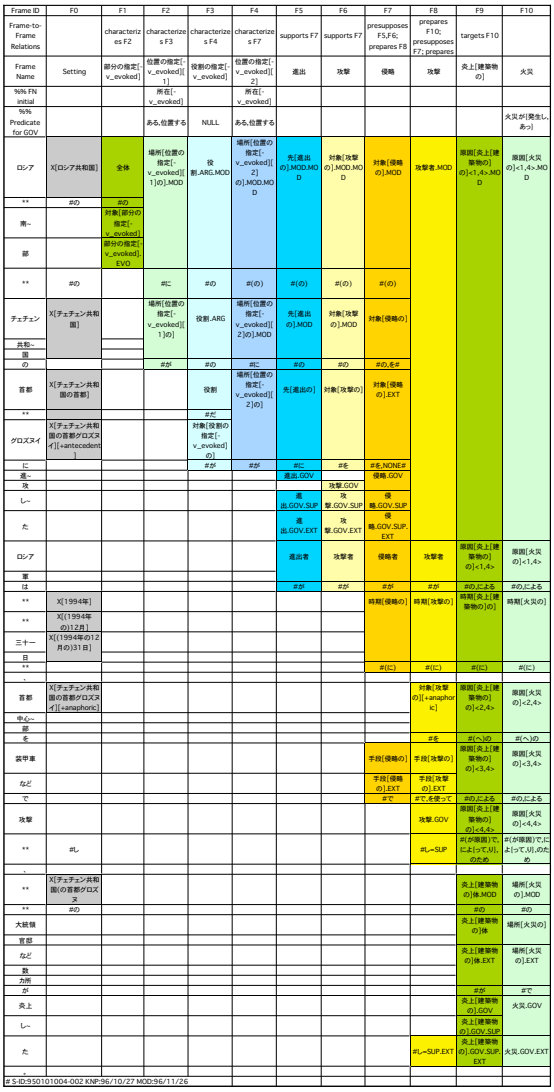

図 1 (7) の MSFA Lite

FrameElement List を使った意味検索は，特定の意味 (役割) タグ *e* (e.g., ⟨対象 [攻撃の]⟩) から部分文字列の集合 (=文書中で *e* の実現値になっている表現の集合) の検索だった．この逆検索の，特定の部分文字列 *t* から意味タグ集合の検索を考えることが可能である．この逆検索は次に述べる配慮から **語句のシソーラス** (thesaurus of words and phrases) を与えるように設計された．

シソーラスは「語」の意味分類体系を与えるものであり[6]，(一部の慣用表現を除いて) それより大きな単位 (例えば句の規模) の意味分類は行なわない．その理由は「語より大きな単位での分類には際限がない」という (必ずしも妥当とは言えない) 直観に基づくものである[7]．

[3] ただ現在公開されている msfa ファイルのすべてこれが順守されているとは言い難い．

[4] *e*.MARKER の値は実際には助詞には限られない．機能動詞=支援動詞や形式名詞であることも少なくない．

[5] Berkeley FrameNet [1] とは違い，私たちの開発したデータベースには，まだフレームとフレーム要素の明示的が定義はない．

[6] 日本語で「語」をどう定義するかという問題は自明な解をもたない問題だが，用言に関しては終止形に代表させ，体言に関しては「の」以外の助詞を含まない要素とするという基準で語を認定できないことはない．

[7] 「句の以上規模の体系分類は，組合わせによって無限に要素が

このため，言語 (例えば日本語) で一般に (a) どんな複合表現が可能か，(b) どんな複合表現がどんな意味をもつかを予測 (あるいは記述) しようとした時，シソーラスの有用性はそれほど高くない[8]．

(a), (b) の問題への理論的に可能なアプローチの一つは，文書集合で有意味な部分文字列を網羅的に列挙するというやり方である．もちろん，これには幾つかの困難が伴う: 第一に，この網羅的列挙が自動的にできる段階にはないため，人手をかけて実行するしかない．その場合，記述対象の (i) 誤認定 (基準の一貫性の欠如を含める) と (ii) 取りこぼしが技術的な問題になるほか，課題達成のための (iii) 所要時間と (iv) 人件費も問題となる．(i), (ii) は人手解析を機械を使った自動処理と較べて場合，本質的弱点だと一般には認識されている．しかし (i) は分類 (に使われるラベル) の誤った排他性に由来する可能性がある[9]．MSFA (Lite) を使ったタグづけは多重のタグづけを許しているので (取りこぼしのリスクは上げつつも) 強すぎる排他性の悪影響をうまく回避しているので，(iii), (iv) の問題を当面度外視するなら，タグづけの作業が継続されたら将来的にそれなりに有用な句のシソーラス (の雛形) を提供する可能性がある．

3 タグづけから派生する言語処理とその含意

3.1 MSFA Lite の記述を利用した文生成

一般に MSFA (Lite) を使って文 s を喚起されている状況 $\sigma_1, \sigma_2, \ldots, \sigma_n$ で記述するということは，s の複合的な意味を基本単位に分解していることに等しい．一般に σ は抽象記述だが，$\sigma_1, \sigma_2, \ldots, \sigma_n$ のおのおのを (s の語彙化情報を参照しつつ) 語彙化する処理 $P : \sigma_i \Rightarrow t_i$ (ただし t は自然言語文 (の集合)) は，(s の意味の分解を経た) 文 (の集合) t への換言/翻訳処理の一種である．

P の実装には，文 s の MSFA で行われる (i) (s 中の要素 x によって) 喚起されるフレーム f の名称 (= f のフレーム名) の指定，(ii) f を構成するフレーム要素 (fe's) の特定とそれらの名称 (=フレーム要素名) の指定，(iii) fe と s の部分文字列の対応関係の指定の三つのほかにも，A. 文の述語の指定，B. fe に後続する (格) 助詞の指定 (これは A の指定に依存する)，C. 文節規模の単位の並べ替えのための情報の三つが必要である．MSFA の元の版は A, B, C の情報を含んでいない．タグづけ速度を上げるために MSFA を簡略化した版である MSFA Lite [6] では A, B, C を実験的に実装し，換言処理を実装した (ただ現状では C は未実装)[10]．

3.1.1 アルゴリズム

基本となるアルゴリズムは次の (8) の通り[11]:

(8)
```
s = "/"
for each fe in fe_array:
  if fe begins_with "#":
     s = s + fe + "/"
  else:
     s = s + "/" + m_array[fe.index]
s = s + "/"
```
ただし fe_array は MSFA の Cj にある (フレームごとの) fe 列，m_array は 1 列目の形態素列とする (fe_array を与える前に要素の並べ替えを行っているなら，後処理なしで C が実装済み)．

単純に言うと，色づけによって，Cj にあるフレーム f フレーム要素として認識されたセル RiCj の値 v が# で始まる (定項の) 場合は v の値が，そうでない (変項の) 場合は形態素列の RiC1 にある値が後続要素となる (ただし v が f.GOV の場合には Predicate for GOV の値が空でない v' なら，RiC1 の値の代わりに v' を値とする)．

3.1.2 生成の見本

この処理によって生成された文の見本を (9) に，その元になった MSFA Lite は図 1 に示したものである:[12]

(9)
a. F1: // ロシア#の// 南 / 部/ NULL=PRED//
b. F2: // ロシア/ **=/ 南 / 部#に// チェチェン/ 共和 / 国#が// ある, 位置する=PRED//
c. F3: // ロシア/ **=/ 南 / 部#の// チェチェン/ 共和 / 国#の// 首都#だ// グロズヌイ#が// NULL=PRED//
d. F4: // ロシア/ **=/ 南 / 部#(の)// チェチェン/ 共和˜/ 国#に// 首都/ **=/ グロズヌイ#が// ある, 位置する=PRED//
e. F5: // ロシア/ **=/ 南˜/ 部#(の)// チェチェン/ 共和˜/ 国#の// 首都/ **=/ グロズヌイ#に// 進出=PRED/ し˜/ た/ ロシア/ 軍#が//
f. F6: // ロシア/ **=/ 南˜/ 部#(の)// チェチェン/ 共和˜/ 国#の// 首都/ **=/ グロズヌイ#を// 攻撃=PRED/ し˜/ た/ ロシア/ 軍#が//
g. F7: // ロシア/ **=/ 南˜/ 部#(の)// チェチェン/ 共和˜/ 国#の, を#// 首都/ **=/ グロズヌイ#を,NONE#// 侵略=PRED/ し˜/ た/ ロシア/ 軍#が// **=X[1994 年]/ **=X[(1994 年の)12 月]/ 三十一/ 日#(に)// 装甲車/ など#で//
h. F8: // ロシア/ **=/ 南˜/ 部/ **=/ チェチェン/ 共和˜/ 国/ の/ 首都/ **=/ グロズヌイ/ に/ 進˜/ 攻/ し˜/ た/ ロシア/ 軍#が// **=X[1994 年]/ **=X[(1994 年の)12 月]/ 三十一/ 日#(に)// 首都/ 中心˜/ 部#を// 装甲車/ など#で, を使って// 攻撃=PRED#し=SUP/#し=SUP.EXT//
i. F10: // ロシア/ **=/ 南˜/ 部/ **=/ チェチェン/ 共和˜/ 国/ の/ 首都/ **=/ グロズヌイ/ に/ 進˜/ 攻/ し˜/ た/ ロシア/ 軍#の, による// **=X[1994 年]/ **=X[(1994 年の)12 月]/ 三十一/ 日#(に)// 首都/ 中心˜/ 部#(へ) の// 装甲車/ など#の, による// 攻撃#(が原因) で, によって, り, のため// **=X[チェチェン共和国 (の首都グロズヌイ)][+anaphoric]#の// 大統領/ 官邸/ など/ 数/ カ所#で// 火災が｛発生し,

---

存在するので，原理的に不可能」と言うのは，ヒトの言語使用の実態には合っていない．ヒトが使う表現は句のレベルでも定型的 (formulaic) である (この傾向は文のレベルでも維持される) [3] (これはコーパス言語学 [2] が明らかにした重要な知見の一つである)．これが事実だとすると，少なくとも句以上のレベルの主要な表現の体系化を行なう努力は決して意味のない努力ではない．可能性をすべて列挙することは原理的に不可能かも知れないが，n 個の語からなる表現集合のうち，一定の代表性のあるものの例えば 70% を網羅することは可能だと思われる．

[8] これは言語表現の非線形性 [4] を考えれば当然のことである

[9] 排他性を想定することは取りこぼし率を下げる効果があるが，取りこぼし率を上げるために必要以上の排他性を要求することは，課題の前提を損なう可能性がある．

[10] 実装は MSFA の作業環境の Excel シート上でそのまま走ることを優先し，Visual Basic for Application (VBA) で行なった．

[11] (9) の見本を生成するコードの細部は (8) と異なっている．

[12] (9) に示したのは，語順の標準化の処理が実装されていない段階での出力である．(9c) では述語 “ロシア/ **=/ 南 / 部#の// チェチェン/ 共和 / 国#の// 首都#だ//” の文末への移動，(9g) では “侵略=PRED/ し˜/ た/” の文末への移動，(9e) と (9f) とで “ロシア/ 軍#が/” の文頭への移動 (か “進出=PRED/ し˜/ た/” と “攻撃=PRED/ し˜/ た/” の前への移動) が必要である．

あっ }=PRED/ た/

*, ** は Setting 列に指定してある情報 “X[...]” にリンクしている (これは文外照応の情報を提供する).

3.1.3 注意

フレームは項と修飾語句の別を決定する (一般的に言って，MSFA では形態素解析が意味解析に先決するものとは考えない)．また形態素とそれらのチャンキングの単位を指定するのもフレームである．(9) に示した見本からわかるように，弱い切れ目と強い切れ目が定義されている[13]．“/” は (形態素境界に相当する) 弱い切れ目，“//” は (文節境界に相当する) 強い切れ目に相当する．従って，MSFA がエンコードしている情報は係り受け解析がエンコードしている情報を包含する．

これらの例では修飾部も生成されているが，文生成の際に (*e*.MOD と指定された行の値の実現を抑制することで) 修飾部のみを選択的に除去することも可能である．これは文意を保存した要約の一種を実現する．

3.2 MSFA ベースの換言処理の含意

ここで紹介した MSFA ベースの換言処理の含意で特に重要だと私たちが思うのは，MSFA に基づいて日本語の言語処理で統語表現の「正解」として扱われている係り受け解析を介しない意味処理が可能であることを示していることである．これが意味することを少しばかり考察してみたい．明白なのは次である: (i) MSFA Lite の一つのフレーム記述は一つの係り受け解析に対応していると考えてよいが，(i) フレーム記述は述語ごとに行われ，(ii) 記述の対象には空の述語を含める必要がある．(ii) 自動化された係り受け解析は KNP や Cabocha という形で実装されているが，MSFA 解析は自動化されておらず，今のところ実装の見こみも立っていない．

(i) と (ii) はおのおの，係り受け解析と MSFA Lite の長所と短所を明確にしている．係り受け解析は表現力不足 (under-representation) であることが (i) で意味される (従って項構造解析などの後処理によって補完されなければならない)．MSFA は十分に実用的でないことが (ii) で意味される．現時点では両者の利点を組み合わせた処理を考えるのがもっとも現実的である．その処理は文 s の解析結果として一般に自動判定された述語ごとに係り受け解析を返す (全体としては (要素の共有箇所を明示した) 係り受け解析の集合を返す) 処理である[14]．

3.3 タグづけ結果の評価

意味タグづけの結果は徐々に蓄積して来たが，その妥当性を評価する手段がなく，その有効性の評価は先送りになっていた．§3.1.1 で概説した分解・換言処理の実装の見通しが立ったので，文 s の意味フレーム $f_1, f_2, \ldots, f_n$ という中間表現へ分解し，それらを機械翻訳した結果 $t_1, t_2, \ldots, t_n$ と，中間表現への分解を行なわない原文 s の機械翻訳の結果 t' とを比較し，妥当性の評価を行なう予定である．

4 終わりに

NICT が 2005 に開始した意味タグづけの開発は，上で紹介した言語資源と言語処理への発展を見こんで行われた．開発の現状は産出される資源の量の点でも質の点でもおそらく NLP 関係者の多くを満足させるものとはなっていないが，それでも従来の技術では未踏な地域に移行するための地図になるくらいの結果は示しているのではないかと考える．

4.1 課題と将来への展望

課題は幾つもあるけれど，もっとも重要だと私たちが認識しているのは．タグづけの終わったデータの有用性の保証である．現在のタグづけは清浄化が終わっておらず，まだ多くの不整合を含む．§3.1 で紹介した換言処理の結果を見て記述にデバッグをかける予定である．

これに続いて将来的に大きな問題となって来るのは，被覆率の向上である．だが，これはタグづけ仕様を改良するなどして効率を上げることのほかに，タグづけの実践者の絶対数が増えることが必要である．今は NICT に所属する 3 人がタグづけを行なっているだけだが，理想的な開発はボランティアベースの open な開発である．これを可能にするよう全部の作業が基本的に Excel 上で行なえるような作業環境を構築することを意識して来た．今後はタグづけ作業に一人でも多くの言語学者がボランティア参入してくれると良いと思っている (私たちはそれが参加者の研究の質の向上に繋がると信じている).

参考文献

[1] C. J. Fillmore, C. R. Johnson, and M. R. L. Petruck. Background to FrameNet. *International Journal of Lexicography*, Vol. 16, No. 3, pp. 235–250, 2003.

[2] J. M. Sinclair. *Corpus, Concordance, Collocation*. Oxford University Press, 1991.

[3] A. Wray. *Formulaic Language and the Lexicon*. Cambridge University Press, Cambridge/New York, 2002.

[4] 池原悟, 阿部さつき, 竹内奈央, 徳久雅人, 村上仁一. 意味的等価変換方式のための重文複文の統語的意味的分類体系について. 情報処理学会研究報告, Vol. 2006-NL-176, pp. 1–8, 2006.

[5] 黒田航, 井佐原均. 意味フレームを用いた知識構造の言語への効果的な結びつけ. 信学技報, Vol. 104 (416), pp. 65–70, 2004. [増補改訂版: http://clsl.hi.h.kyoto-u.ac.jp/~kkuroda/papers/linking-l-to-k-v3.pdf].

[6] 黒田航, 李在鎬. MSFA Lite を使った意味タグづけの仕様, 2007. http://clsl.hi.h.kyoto-u.ac.jp/~kkuroda/papers/msfa-lite-spec.pdf.

[7] 黒田航, 飯田龍. 文中の複数の語の (共) 項構造の同時的, 並列的表現法: Pattern Matching Analysis (Simplified) の観点からの「係り受け」概念の拡張. 信学技法, Vol. 106, No. 191, pp. 1–5, 2006.

[8] 内山将夫, 高橋真弓. 日英対訳文対応付けデータ. http://www2.nict.go.jp/x/x161/members/mutiyama/align/index.html, 2003.

[9] 飯田龍, 小町守, 乾健太郎, 松本裕治. 日本語書き言葉を対象とした述語項構造と共参照関係のアノテーション: Naist テキストコーパス開発の経験から. 言語処理学会第 13 回年次大会発表論文集, 2007.

[13] これは (8) の “/” の挿入のタイミングと場所で実装.

[14] SynCha (http://cl.naist.jp/~ryu-i/syncha/) [9] がこれに近い処理を目指していると思われる．(Parallel) Pattern Matching Analysis (PMA) [7] は (処理系としては未実装だが) ここで想定している処理をモデル化していると思われる.