

今の機械翻訳に利用者が望めること，望めないこと

『日本語学』の機械翻訳特集への寄稿

黒田 航 加藤 鉦三

2009/08/26, 27, 28, 30, 31; 09/01, 02

1 はじめに

1.1 機械翻訳の進歩はなぜ異例に遅いか

電子計算機による自動翻訳（以下では単に機械翻訳と呼ぶ）の歴史は古い。ENIAC が誕生した 1946 年時ですで見こみのありそうな応用の一つに機械翻訳が挙げられており，翌 1947 年の W. Weaver と A. D. Booth による機械翻訳実験初期のシステム [Booth & Booth 2000] は，機械翻訳が（暗号解読を模倣することで）原理的に実現可能であることを示した（最初期の試作システムは簡単な幾つかの露文を英訳できたが，実用的なものではなかった）。それから 60 年近く経ち，計算機の処理能力が，処理速度の面でも記憶容量の面でもケタ違いに高まった今でも，自動翻訳は大いに未完成な技術の一つである。

計算機の処理能力の向上，活用可能な資源の良質化に伴って，機械翻訳の性能は確かに不断に改善されて来ている。だが，機械翻訳の歴史の古さを考えると，その改善の速度の遅さは異常なほどである。それよりずっと後発で

ずっと短時間に実用可能になったシステムは幾つも存在する (OCR やカーナビゲーションやインターネットはその好例である) . この違いは非常に対照的であり , かつ示唆的であるように思える .

その理由が , 計算資源が今だに実用的な機械翻訳に必要なレベルに達していないからなのか , それとも機械翻訳の理論が最初期からまったく進歩していないせいなのかはわからない . ここでは , 90 年代になって大流行した統計ベースの翻訳は , 最初期のモデルを妥協なしで実装したものに近く , (統計上のトリックは別にすると) 技術的な革新は特になかった点を理解しておくことは有用である (後述するが , 90 年代以来の統計翻訳の大流行は , 機械翻訳の伝統では「先祖帰り」だった) . 仮に理由が翻訳の理論の旧態依然だとすると , その機械翻訳の進歩の異例な遅さを言語処理研究者だけに求めるのは不当である . 彼らの仕事は言語の本質 (というものが仮にあるとするならば , それ) の解明ではないからである .

とはいえ , 機械翻訳の実用的なシステムを実現しようとする努力の中で工学系の研究者が , 言語学者よりも深い言語の実態の理解を達成した可能性はある . 例えば池原ら [Ikehara 2001, 池原 2004a, 池原 2004b] の非線形性を想定した言語処理はそういうものだ と評価することは十分に異義のあることだろう . ただ , そのような考えを体現したシステムでも , まだ十分な性能は示していない . 実際のところ , それほど機械翻訳は難しい . この点には言語学者も「対岸の火事」を決めこまず , 十分な理解を示す必要がある . 機械翻訳を含めた言語処理の進歩が遅いのは , 言語処理研究者の無能や理解不足が理由だと考えているのであれば , それは根本的な事実誤認である , ヒトの使っている言語の実態は非常に複雑 — 少なくとも言語学者が知っている以上に複雑 — であるのは確実であり , その複雑性をうまく表わせるモデルが存在していないことが言語処理の進歩の遅さの本当の原因だという可能性の方がずっと大きいからである . 言語学者は言語の複雑度を明らかに軽く見積る傾向があるので , これは強調しておきたい . 言語の記述はいわゆる「文法」を書き下せばそれで終わりではない (どちらかという と , 言語学者は言語処理

によって「有効でない」と判定されたモデルを擁護する傾向がある)。

1.2 機械翻訳モデルの二つの極

機械翻訳を実装するモデルには二つの対立極がある。文法ベースのシステム (grammar-based systems: GBS) と統計ベースのシステム (statistically based systems: SBS) である。GBS は必要な資源の開発を手で行なう。資源は辞書と文法に分れる。SBS は辞書と文法を開発を機械に任せ、人手の介入をゼロに近づける。近年の SBS の進歩は 90 年代に起こった統計手法の高度化と関係している。二つの枠組みは根本的なところで異なる。まずそれを説明しよう。

ヒトが行なっている翻訳 (HT) との関係から見て、機械翻訳 (MT) のシステムを実装する方針は次の二つに大別できる:

- (1) a. HT と (ほぼ) 同じタイプの処理を計算機上で実装している保証はないが、(理由はともかく) HT と (ほぼ) 同じ出力を与える MT システムを作る
- b. HT と (ほぼ) 同じタイプの処理を計算機上で実装し、HT と (ほぼ) 同じ出力を与える MT システムを作る

ここで次の点には気を配っておいた方がよい: GBS の目標は (1b) であるが、§2 で述べる理由で、明らかに不十分である。それは単に GBS では開発経費が対価に見合わないという理由ばかりでなく、モデルが誤っているからだという可能性がある。例を挙げるなら、おそらくヒトの言語知識が「文法」と呼ばれる規則の体系で特徴づけられる度合いより、膨大な事例記憶に依存している度合いは、かつて考えられているよりずっと大きいに違いない [黒田 2007, 黒田・長谷部 2007] (これは本稿では取り上げないが翻訳メモリー (translation memories) と呼ばれるシステムの効用にも現われている)。

だが、GBS のモデルが誤っていることは、その対局にあると思われる SBS

の妥当性を保証するか? おそらくそうではない。SBS の目標は (1a) であって、(1b) ではないからである。第一に、(1a) の実現は (1b) の実現よりも簡単であるという理由で、そうであり、第二に、SBS の起源が暗号解読である点から見ても、そうである。ヒトの行なう翻訳は「単なる」統計処理や「単なる」暗号解読でないことは确实だからである(そうでなければ翻訳メモリーのような冗長なシステムに出番はないだろう)。

ここでは (1a) の目標の達成可能性に関して、次の問題があることは理解しておいた方がよいだろう:

- (2) HT と (ほぼ) 同じタイプの処理を計算機上で実装しないで、HT と (ほぼ) 同じ出力を与える MT システムを作るとは、「理論的に」ではなく「技術的に」可能か?

SBS を真に受けるならば、(2) への答えは肯定的である。だが、そのような答えを出す際、人は暗黙の制限を設けていないか? 例えば「十分に多くのデータがあれば」と良く言われるが、必要なデータが例えばヒトの個人が一生に聞いたり読んだりする言語表現の量の数十倍の量のデータなのであれば、それは何かが根本的におかしいと言うべきである。少なくとも、その量が本当に「技術的に可能」な範囲に収まっているのか、期待から独立した答えを出せるほど公平無私な研究者がいるとは思えない。

(2) への答えが否定的だとすると、説明される必要があるのは「なぜ GBS が期待されていたほどうまくいかなかったか?」である。それは MT を実装するシステムの妥当な候補は、単純な GBS でもなく、かといって単純な SBS でもないようなシステムだということを示唆することになる。

この点を明確にするため、§2 で機械翻訳の動作原理を説明し、それに続く §3 で現状の機械翻訳の問題点の一つを取り上げ、言語学の立場から改善策を提案する。

2 機械翻訳システムの分類と傾向

GBS と SBS とは機械翻訳への質的に異なったアプローチであり，必要となる資源のタイプが異なる．それを明らかにするために，翻訳の原理を簡単に説明しておくことにする．

(機械) 翻訳の基本原則 元言語 S の表現 s を先言語 T の表現 t に翻訳するという課題は，定義としては次のように簡単である：

- (3) a. 言語 S の表現 s が表わす意味 m を求める．
- b. 言語 T の (有意味な) 表現 $T = \{t_1, t_2, \dots, t_n\}$ を定める．
- c. T のうちでもっとも m に近い意味をもつ t_i を一つ選ぶ．

以上の処理が s の t への翻訳である．

- (4) (3a)–(3c) のすべての処理を人手で行なった場合，それは人手翻訳であり，それを機械が行なった場合，それは機械翻訳である．

GBS の基本的な仕組みは次である：

- (5) 前処理として次のデータベース構築を行なう：

- a. S, T のおのこのについて，辞書 $L(S), L(T)$ と文法 $G(S), G(T)$ を人手で開発する．
- b. S, T の語の (意味の) 対応づけデータ $A(w_s, w_t)$ ($w_s \in L(S), w_t \in L(T)$) を人手で構築しておく．この際，
 - i. s_i と t_i を中間表現 m を使って対応づける場合と， m を使わないで対応づける場合がある (前者は次元を圧縮する効果があり，一般に必要なデータは後者の方が大きくなる)．
 - ii. 語よりも大きな非構成的な意味をもつ「非線型」な複語単位 (nonlinear/multiword expressions) [池原 2004a, 池原 2004b] の対応づけも別に用意しておく必要がある (語よりも大きな

単位の意味の翻訳を優先しないと (言語の意味の非構成性/非線形性により) 翻訳の精度が下がることが知られている) .

(6) $s \Rightarrow t$ の本処理は次のように行なう:

- a. s を $L(S)$ と $G(S)$ を使って構文解析 (parse) する . その結果を p とし , p の意味表示を m とする .
- b. p か m を基にして $A(S,T)$ を使って T の語集合 w を得る .
- c. $G(T)$ を使って w から t を構築する .

SBS の典型的なやり方は次である:

(7) 前処理として次のデータベース構築を行なう:

- a. S, T の一方から他方への翻訳対 $A(S,T) = \{ (s_1, t_1), (s_2, t_2), \dots, (s_N, t_N) \}$ を大量にデータとして用意する ($A(S,T)$ を S と T の対訳コーパス (parallel corpus) あるいは対応づけデータ (alignment data) と言う) .
- b. $A(S,T)$ の部分文字列 (e.g., 単語) の分布がもつ統計情報を利用して , 文レベルの対応を句や語レベルの対応関係 $P(S,T)$ や $W(S,T)$ に分解する . $W(S,T)$ は対訳辞書 , $P(S,T)$ はフレーズ表 (phrase table) と呼ばれる .

(8) $s \Rightarrow t$ の本処理は次のように行なう:

- a. 既製のデータベースである $P(S,T)$ と $W(S,T)$ から S 側が s の部分文字列に一致する $p(\in P)$ と $w(\in W)$ を選ぶ .
- b. p の T 側と w の T 側を関数 Φ で合成して $t = \Phi(p,w)$ を得る .

GBS と SBS の一長一短 歴史的に見ると SBS が GBS より古い , 50 年代後半からの生成文法の大流行により SBS はほぼ完全に GBS に取って代わられた . 80 年代後半に始まる SBS の再評価とそれに続く大流行は , 最初期の Weaver-Booth 流の機械翻訳モデルの再興である .

SBS は実用化の前提となる計算資源や言語資源の量が GBS に較べてケタ

違いに大きい点は理解しておくべきである。この再興が可能になった背景は、次の三つが同時期に重なった点にある: (A) GBS は拳動の脆さや被覆率の低さという弱点を克服できずに低迷していた。(B) 90 年代に入って計算機の処理能力が大きく改善された (初めてメガバイト級の演算速度が一般機上で実現され、初めてギガバイト規模のデータを扱えるようになった)。(C) (おそらく (B) の副産物であるインターネットの急速な普及により) 言語資源の供給不足の問題が部分的に克服された。

(A) についてももう少し考察しよう。過去の経験から、GBS には次の難点があることがわかっている:

- (9) a. 計算資源 —特に $G(S)$, $G(T)$, $A(S, T)$ — の開発に相当の時間と経費がかかる (非効率性の問題)
- b. 未知語に対応できない ($L(S)$ の被覆率の限界の問題)
- c. 入力が破格文である場合にうまく動作しない ($G(S)$ の被覆率の限界の問題)
- d. T で「自然」な訳が得られないことが多い (翻訳単位を決めている $A(S, T)$ の非最適性の問題)。

GBS で問題になった (9a), (9b), (9c), (9d) は SBS では合理的な形で解消されるが、すべてがうまく行くわけではない。SBS では、(9a) で問題だった文法と辞書の手構築するという手間も費用もかかる前処理を回避できる。(9b) の問題の一部は、与えるデータの規模を大きくすれば自然に解消されると期待される。だが、GBS で記述できる規則性のすべてが SBS で同じ精度で記述できるわけではない。SBS では n gram の窓の大きさに制限があるため、GBS なら簡単に記述できる長距離依存が捉えられないことが少なくない。つまり、与えるデータの規模が十分に大きいとしても、それから自動獲得される規則の被覆率が十分になるという保証はない。

構文解析を前提としない SBS は原理的に (9c) の問題とは無縁でいられる。更に言うと、精度低下の原因となる統語解析を回避し、かつ (3a) の m

を求める処理も回避できる。これは GBS に対する (工学的に見る限り) 大きな利点である。だが、GBS が抱えていた問題 (9d) の改善について、SBS の性能は両義的である。SBS の出力は翻訳がうまく行った場合と行かなかった場合の落差が非常に大きい。うまく行かなかった場合 — しかも、それは決して稀ではないのだが — SBS の出力は有意義な表現どころか、容認可能な表現にすらならない。

EBS という選択肢 当然のように頭に浮かぶのは、GBS と SBS の長所を合わせたシステムは作れないのか? という疑問である。だが、それは (1) で述べた目標の違いにより、簡単には達成できない。SBS は結果の同一性だけを求め、人手翻訳と機械翻訳の処理の同型性を追求しない。これに対し、GBS はそれより更に高度な想定をもつ。だが、その対価はしばしば開発経費に見合わない。とはいえ、50 年代後半から生成文法の普及によって機械翻訳の主流は変わったが、その理由は端的に言えば、対訳データを作るより、辞書と文法を用意する方が安上がりだったからだということも思いだす必要もある (工学の分野には、なるべく費用対効果が高いものを作ろうとする根強い (おそらくそれなりに健全な) バイアスがある)。

アナロジーに基づく機械翻訳 [佐藤 1997] の発展形として普及している用例/実例ベースのシステム (example-based systems: EBS) [Hutchins 2005] は、GBS と SBS の中間的なものである。現状の EBS は統計ベースの処理モデルでありながら、同時に (1b) を目指していると思わせる (EBS は統計ベースで実装されることが多いが、これは不可避ではない)。これは言語表現の定型性 [Wray 2002] の重要性を考えると、おそらくもっとも期待できる機械翻訳の方向性であり、精力的に研究が進められているが、それでも十分な性能は実現していない。その理由の一つは、入力表現に部分一致する実例が複数ある時に、それらを選択的に統合する処理の効率的なアルゴリズムが開発されていない点にある。

今の機械翻訳に望めること、望めないこと 以上のことから、現状の機械翻訳に何が望めて、何が望めないかは見当がつくと思う。動作原理の違いに

起因する次のような一長一短がある:

- (10) a. SBS/EBS は (GBS より) 未知語や新語の処理に強い .
- b. GBS は定型・慣用表現に弱い , SBS/EBS は定型・慣用表現に強い .
- c. SBS/EBS の出力は当たり外れが大きいが , 大きく当たることもある . GBS の出力は当たり外れが小さいが , 大当たりは滅多になく , 全体的として見ると満足に行かない訳が多くなる (だが , 平均を較べると , 一方が他方よりいいとは言いがたい) .
- d. GBS と SBS/EBS とでは違うタイプの誤訳が出る . 例えば SBS/EBS では長距離呼応関係になる否定要素はちゃんと訳されないことがある (タイプが異なるので , どちらが誤訳率が高いというのは言いにくい) .

SBS/EBS で根本的に問題なのは , (少なくとも現時点では) $t = \Phi(p, w)$ が T の表現であることを保証するように関数 Φ を設計する方法が知られていないという点である . これの意味するのは , ((3b) で問題になるように) t が T の要素であるように出力を制限しない場合 , システムの出力は有意味な表現の集合である T に属さない可能性があるということである .

この問題の解決は簡単ではない . t が T の要素であるように出力フィルターで制限をかけるという案が考えられるが , それでシステム出力がゼロ個になってしまう可能性は少なくない . それ以前に , 出力フィルターが自動学習できるならばよいが , それを人手開発する必要があるとすれば , SBS の GBS に対する長所の一つは大きく損なわれる . $G(T)$ を自動獲得してそれを T 側の出力フィルターとして使う技術があればよいのだが , それが可能であれば , 少なくとも T 側の出力フィルターの実装が $G(T)$ (の簡易版) の開発を含むのは不可避である . これは翻訳ソフトを使う人と作る人の思惑の違いがハッキリする点の一つである . 以下ではそれを明らかにする .

3 翻訳ソフトの開発者と利用者のすれちがい

機械翻訳の利用者にとって、最も困ることは、訳文の意味が分からない、という事態であろう。例えば (11) の原文の翻訳結果が (12) のようなものであれば、それは使い物にはならない。

(11) The market does what the market wants to do, not what the Fed, Mr. Buffett, or anyone else thinks it should do (more about that later).

(12) 市場は、市場が行いたいことを行います、ない、連邦準備局、バフェット氏あるいは誰か他の人が思うこと、それが行うべきです（後でそれに関してより多く）。[The 翻訳 2007 プレミアム]

この事例では、翻訳エンジンが原文の構文解析に失敗している。よって、利用者が開発者に最も強く望むのは (13) であろう。

(13) 原文の構文解析の精度向上

(13) については、利用者側と開発者側とで認識を共有できているはずである。よって、本稿では (13) は問題にしない。以下、本稿では、開発者側があまり意識していないと思われる利用者側の潜在的な要望を概観したい。

ネット上の無料の機械翻訳でも、また市販の機械翻訳ソフトでも、訳語を利用者が選択できるという機能がある。例えば (14) の原文に対し、(15) の出力が得られるが、

(14) As investors, we like to receive affirmation for our decisions.

(15) 投資家として、我々は我々の決定のために断言を受けることを好みます。[LogoVista 2009 PRO]

利用者の方で、例えば、「どうもこの『断言』は違うのではないか？」と疑問を感じた場合、翻訳ソフトの編集画面上で、原文の affirmation、もしくは

訳文の「断言」をクリックすると、別の訳語候補が表示されるため、そこで最適な訳語を選択することができる。

実は(15)は誤訳であり、「決定に対する確証」が正しい解釈である。ここで問題にしたいのは、利用者にとって、その解釈が到達可能であるかどうか、である。先の affirmation と同様に、for をクリックすれば、別の可能性が表示される。だが、その選択肢は 27 もあるため、利用者にとっては、負担にはなっても助けにはならないというのが現実である。これは英語を得意とする者にとってもそうなのであるから、英語の不得意な者にとっては利用価値はないに等しい。

(14) の for 句の解釈は、訳語の問題というよりは、実は構文解析の問題であるとする方が、より効率的かつ現実的である。つまり、(14) の for 句の解釈は、いきなり 27 の可能性を考えるのではなく、まずは(16)の二つの可能性から考えるべきである。

(16) for 句は、affirmation を修飾する連体要素か、receive を修飾する連用要素のいずれかである

(16) は、for という単語レベルではなく、for 句というフレーズレベルにおいて、それがどこに位置しているのか、を問題にしたものである。さて、機械翻訳ソフトは、訳文を出力する際、程度の差こそあれ、原文の構文解析という作業は必ず経ている。(16)に関して言えば、この選択を翻訳ソフトが失敗した、ということである。

利用者が入力表現の構造を指定するオプション さて、上記のような、単語レベルの別訳候補を表示する機能に加えて、本稿では(17)を提言したい:

(17) 構文解析レベルでの別解釈候補を表示する機能を実装する

(17) は、(14) の場合、(15) の結果と並んで、利用者の求めがあれば、「for 句は affirmation を修飾する」という、別の構文解析候補を表示するようにしたらどうか、ということである。もしそのような構文解析情報が与えられた

としたら，利用者が「～のために」という誤訳を捨て，「～に対しての」という正訳を選択できる可能性は圧倒的に高まることが期待される．要点は，入力文の構文解析のバイアスを人が与えるということである．機械翻訳の現状では入力文の解析への人手の介入はゼロである．これを制限を緩めれば機械翻訳はもっと使えるようになるのではないか？ということである．

(17) の提案のミソは現実性である．機械翻訳エンジンは，原文が与えられると，ほとんど無限大の構文解釈の中から最終的に一つの解釈結果に絞り込む，という作業をしている．(17) の提案は，その絞り込み作業の中の最後の方の段階を，利用者にも見えるようにしたらどうか，というものであるため，翻訳エンジンに対し新しい作業を要求するものではない．

もう一つ，(17) の提案には大きな理由がある．現状の機械翻訳においては，統語的な誤りのほとんどが，ある要素が何を修飾しているのか，に関するものである．(17) の提案は，例えば (15) の二つの解釈のうちどちらを選択するか，というとても難しい課題を機械にやらせるのではなく，ある意味，機械よりはるかに処理能力の高い人間，すなわち利用者，にやらせよう，ということを行っている．いわゆる「文脈から判断する」という作業は，現状では，機械より人間の方がはるかに上手にやる．それを使わない手はないのではないだろうか．それを使うためには仕掛けが必要であり，それが (17) である．

機械翻訳の「利用者」とは誰か　ここでは機械翻訳の「利用者」の用途が一樣だと事態を単純化している．それが事実でないのは明らかであり，それは別の次元での問題を生じさせる．日英間の翻訳を例に考えよう．第一の次元に，翻訳の方向性の軸 ($J \Rightarrow E$ の翻訳と $E \Rightarrow J$ の翻訳の区別) があり，第二の次元に，利用者の英語に関する知識レベルがある (話を単純化するために，日本語の知識レベルの差は無視できるとしておく)．単純化によって，英語の知識レベルを次の三つの段階に分けることにすると，表 1 のように六つの需要のタイプがある (日本語のレベルも三段階に分けるなら， $18 (=2 \times 3 \times 3)$ 個のタイプがある)：

表 1 英語のレベルと翻訳の方向の組合わせ

英語の知識	$E \Rightarrow J$	$J \Rightarrow E$
高	$H(E \Rightarrow J)$	$H(J \Rightarrow E)$
中	$M(E \Rightarrow J)$	$M(J \Rightarrow E)$
低	$L(E \Rightarrow J)$	$L(J \Rightarrow E)$

これらの異なるタイプの利用者は確実に異なるタイプの要求をもっている。そのため、単一の MT システムで、これらのすべてのタイプの利用者の要求を満足させることはおそらく不可能である。開発者もこれを承知し、特定の利用のタイプを想定したシステムを開発していると考えられるが、それが利用者から見えるようにはなっていないようだ。多くの開発者は、利用者の潜在的な要求を「決め打ち」しているように見える。これが正しいのであれば、翻訳ソフトと開発者と利用者の最大の「すれちがい」は、開発者が自分の開発した翻訳ソフトが表 1 のどのタイプの利用者を想定しているかを明言していないために、利用者は自分の用途にあったシステムを効率的に選べない点にある。

レベルの異なる複数の翻訳エンジンからの選択 今の技術レベルでは、表 1 のタイプごとに実装すべき処理が異なっている可能性がある。それらの個別システムが独立に実装されたという想定の下で利用者にとってもっとも有益だと思われるのは、利用者に自分のにとって有益な個別システムを選択させるという方法かも知れない。

$H(E \Rightarrow J)$, $H(J \Rightarrow E)$, $M(E \Rightarrow J)$, $M(J \Rightarrow E)$ の利用者の視点に限って言うと、もう一つ利用者の視点からあってもよいと思うのは、翻訳エンジンの選択可能性のオプションである。規則ベースのエンジンの出力と統計ベースのエンジンの出力の両方を表示するというのは、利用者の側からすると悪くない選択肢であるように思う。

4 終わりに: 機械翻訳の今後

開発費用が資金的にも時間的にも GBS に較べて激減する点は、工学者から見た SBS の最大の利点である (更にもう一点、SBS は人手誤りを回避できると言われることもある。だが、この評価は無条件ではない。自動解析が人手処理を性能的に上回るのは、形態素解析や品詞タグづけのような比較的、低次の処理だけである)。

SBS の最大の難点は出力 t の当たり外れの大きさである。これは EBS でも劇的に改善されているとは言い難い。その理由は第一に、(7b) で適当な $P(S, T)$ を見つけることが $t = \Phi(p, w)$ が常に T の要素になる性質をもたせることから独立していないという点が、SBS/EBS で考慮されていないためである。逆に言えば、 $P(S, T)$ の要素として、後でうまく合成可能な要素を獲得できれば、問題は一気に解決するようにも思える。もちろん、どんな要素が「後でうまく合成可能な要素」なのかは今の時点で確立していないが、それが変項の位置情報と意味情報を保存したパターン/非線型表現であることは、ほぼまちがいないだろう。また、複数のパターンの合成の動作原理は、パターンへの語彙的要素の代入ではなく、([Kuroda 2001, 黒田・飯田 2006] が特徴づけるような) パターンの同士の重ね合わせであると考えられるのも、おそらく確実である。

参考文献

- [Booth & Booth 2000] A. D. Booth and K. H. V. Booth. The beginning of mt. In W. J. Hutchings, editor, *Early Years in Machine Translation*, pp. 252–261. John Benjamins, 2000.
- [Hutchins 2005] J. Hutchins. Example-based machine translation: A review and commentary. *Machine Translation*, 19:197–211, 2005.

- [Ikehara 2001] S. Ikehara. Meaning comprehension using semantic patterns in a large scale knowledge-base. In *Proceedings of the PACLING'01*, pp. 26–35, 2001.
- [Kuroda 2001] K. Kuroda. Presenting the PATTERN MATCHING ANALYSIS, a framework proposed for the realistic description of natural language syntax. *Journal of English Linguistic Society*, 17:71–80, 2001.
- [Wray 2002] A. Wray. *Formulaic Language and the Lexicon*. Cambridge University Press, Cambridge/New York, 2002.
- [池原 2004a] 池原 悟. 非線形な言語表現と文型パターンによる意味の記述. 情報処理学会研究報告, 自然言語処理研究会, 2004-NL-159:139–146, 2004.
- [池原 2004b] 池原 悟. 自然言語処理と言語過程説. In 佐良木 昌, editor, 言語過程説の探求 第 1 巻: 時枝学説の継承と三浦理論の展開, pp. 333–408. 明石書店, 2004.
- [黒田 2007] 黒田 航. 徹底した用法基盤主義の下での文法獲得: 「極端に豊かな事例記憶」の仮説で描く新しい筋書き. 月刊言語, 36(11):24–34, 2007.
- [黒田・長谷部 2007] 黒田 航・長谷部 陽一郎. Pattern Lattice を使った (ヒトの) 言語知識と処理のモデル化. In 言語処理学会第 15 回大会発表論文集, pp. 670–673, 2009.
- [黒田・飯田 2006] 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, 106(191):1–5, 2006.
- [佐藤 1997] 佐藤 理史. アナロジーによる機械翻訳. 共立出版, 1997.