マッチング・ファンド方式による産学連携研究開発事業

# 複数言語にまたがる言語知識処理技術の研究

研究開発プロジェクト総括研究成果報告書

平成 13 年 5 月


総括代表者　田中 穂積
東京工業大学 大学院情報理工学研究科 教授
企業分担代表者　亀井真一郎
日本電気株式会社 情報通信メディア研究本部 主任研究員
企業分担代表者　梶 博行
株式会社日立製作所 中央研究所
マルチメディアシステム研究部 主任研究員
企業分担代表者　松井くにお
富士通株式会社 ＤＢサービス部 担当部長
企業分担代表者　平川 秀樹
株式会社東芝 研究開発センター
ヒューマンインターフェースラボラトリー室長

## はしがき

インターネットの爆発的普及により，複数言語にまたがる言語処理応用システム (機械翻訳，多言語情報検索など) はますますその重要性を増している．本研究では大量のコーパスから言語処理技術に役立つ知識を抽出し，これらの応用システムを高度化する技術の研究開発を目指して研究をおこなってきた．このために，大学側の研究成果である言語知識獲得技術を大規模なコーパスに適用し，得られた知識を企業側の持つ機械翻訳システムや多言語情報検索システムに適用して，システムの高度化をはかった．それと同時に，大学側の基礎研究成果を実用システムに利用することによって，研究成果の洗練をおこなった．

東京工業大学の田中・徳永の研究グループは形態素統語解析ツール MSLR パーザの開発と情報検索における検索質問拡張に関する研究を中心に研究をおこなった．MSLR パーザは従来から東京工業大学のグループで開発を続けてきたもので，本研究ではこれまでの研究成果を集大成し，誰でも使えるような形にツールキットとしてまとめあげた．このツールは一般に公開され，すでに多くの研究グループによって使用されている．研究内容の詳細については，添付の論文「自然言語解析のための MSLR パーザ・ツールキット」を参照されたい．また，このツールを評価する目的で 10,000 例文に統語構造を付与したコーパスを作成した．

検索質問拡張に関する研究では，これまでにも多くの研究がおこなわれているが，東京工業大学のグループでは性質の異なる複数のシソーラスを組み合わせることによってそれぞれのシソーラスの欠点を互いに補完する手法を開発した．大規模なデータを用いてこの手法を評価し，その有効性を確認するとともに，手法の限界についても明らかにし，その解決策について考察をおこなった．研究内容の詳細については，添付の論文「The exploration and analysis of using multiple thesaurus types for query expansion in information retrieval」を参照されたい．

機械翻訳における最大の課題は，源言語の解析のレベルをあげることと，適切な訳語選択の方式を考えることである．京都大学では，この 2 つの研究課題にとりくんだ．前者については，京都大学で開発してきた形態素・構文解析システムを整備し，産業界で利用可能な形にした．さらに，大量のコーパスから格フレーム辞書を自動的に構築する方法を考案し，この辞書に基づく格解析，文脈解析のシステムを構築した．この研究内容については，添付の論文「用言の直前の格要素の組を単位とする格フレームの自動獲得」を参照されたい．後者の問題については，日本語単語の典型的な用法をコーパスの KWIC などを参照して列挙し，それらを人手で翻訳することにより「翻訳メモリ」を構築した．この翻訳メモリを用いて訳語選択の世界規模のコンテストを主催し，活発な議論を行うとともに，翻訳メモリの有効性を検証した．

東京大学のグループは，HPSG の枠組で記述された文法を用いる頑健で高速なパーザを開発してきた．本研究では，これまでの基礎研究の成果をまとめ，実用に耐えうるパーザを実現すると同時に，大規模な文法を開発し，この文法を用いて評価実験をおこなった．

この研究の詳細については，添付の論文「The LiLFes abstract machine and its evaluation with the LinGO grammar」を参照されたい．このパーザの開発と平行して，解析されたテキストから対象分野のオントロジーを構築するためのツールを開発し，分子生物学の分野のテキストについてコーパスを作成した．

## 研究組織

| | | |
|---|---|---|
| 総括代表者 | 田中穂積 | (東京工業大学・大学院情報理工学研究科・教授) |
| 研究分担者 | 辻井潤一 | (東京大学・大学院理学系研究科・教授) |
| 〃 | 徳永健伸 | (東京工業大学・大学院情報理工学研究科・助教授) |
| 〃 | 黒橋禎夫 | (京都大学・大学院情報学研究科・講師) |

## 研究経費

95,700 千円

## 研究成果・発表など

- Kentaro Inui, Virach Sornlertlamvanich, Hozumi Tanaka, Takenobu Tokunaga. Probabilistic GLR Parsing. Advances in Probabilistic and Other Parsing Technologies, Kluwer Academic Publisher, Chapter 5, pp. 85-104, 2000, Dec.
- 白井清昭, 植木正裕, 橋本泰一, 徳永健伸, 田中 穂積. 自然言語解析のための MSLR パーザ・ツールキット. 自然言語処理, Vol. 7, No. 5, pp. 93-112, 2000, Nov.
- 徳永健伸. 結合価情報を用いた語順の推定. 月刊「言語」, Vol. 29, No. 9, pp.68-75, 2000, Aug.
- Timothy Baldwin, Hozumi Tanaka. The Effects of Word Order and Segmentation on Translation Retrieval Performance. Proceedings of the International Conference on Computational Linguistics, pp. 35-41, 2000, Jul.
- 田中穂積, 亀井真一郎, 森口稔, 加藤安彦. 大きなコーパスを共有しよう. 情報処理, Vol. 41, No. 7, pp. 774-786, 2000, Jul.
- 松本裕治, 徳永健伸. コーパスに基づく自然言語処理の限界と展望. 情報処理, Vol. 41, No. 7, pp. 793-796, 2000, Jul.
- 橋本泰一, 白井清昭, 徳永健伸, 田中穂積. 統計的手法に基づく形容詞または形容動詞の修飾先の決定. 情報処理学会自然言語処理研究会, Vol. 2000, No. 65, pp. 87-94, 2000, Jul.
- Kiyoaki Shirai, Hozumi Tanaka, Takenobu Tokunaga. Semi-Automatic Construction of a Tree-Annotated Corpus Using an Iterative Learning Statistical Language Model. Second International Conference on Language Resources and Evaluation, pp. 461-466, 2000, May.
- Rila Mandala, Takenobu Tokunaga, Hozumi Tanaka. Query expansion using heterogeneous thesauri. *Information Processing and Management*, Vol.36, No.3, pp.361-378, 2000, May.
- Tokunaga Takenobu, Ogibayashi Hironori and Tanaka Hozumi. Effectiveness of complex index terms in information retrieval. The 6th RIAO Conference (RIAO 2000), pp. 1322-1331, 2000, Apr.
- Rila Mandala, Takenobu Tokunaga, Hozumi Tanaka. The exploration and analysis of using multiple thesaurus types for query expansion in information retrieval. 自然言語処理, Vol.7, No.2, pp.117-140, 2000, Apr.

- 徳永健伸. 言語処理は情報検索に役立つか？. 日本音響学会 2000 年春季研究発表会講演論文集, pp. 31-32, 2000, Mar.
- 白井清昭, 徳永健伸, 田中穂積. 単語の共起データを用いた構文的な統計情報の学習に関する研究. 言語処理学会第 6 回年次大会, pp. 151-154, 2000, Mar.
- 荻林裕憲, 徳永健伸, 田中穂積. 情報検索における索引語の選択的利用. 言語処理学会第 6 回年次大会, pp. 439-442, 2000, Mar.
- Timothy Baldwin and Hozumi Tanaka. Verb alternations and Japanese — How, what and where? *Proceedings of the 14th Pacific Asia Conference on Language, Information and Computation (PACLIC 14)*. pp. 3-14, 2000, Jan.
- Yusuke Miyao, Takaki Makino, Kentaro Torisawa, and Jun-ichi Tsujii The LiLFeS abstract machine and its evaluation with the LinGO grammar, Journal of Natural Language Engineering, Cambridge University Press, Vol 6(1), pp. 47–61, 2000.
- Kentaro Torisawa, Kenji Nishida, Yusuke Miyao, and Jun-ichi Tsujii An HPSG Parser with CFG Filtering, Journal of Natural Language Engineering, Cambridge University Press, Vol 6(1), pp. 63–80, 2000.
- 金山博, 鳥澤健太郎, 光石豊, 辻井潤一, 3 つ以下の候補から係り先を選択する係り受け解析モデル, Journal of Natural Language Processing (言語処理学会学会誌) Vol. 7(5), 2000.
- Hiroshi Kanayama, Kentaro Torisawa, MITSUISHI Yutaka, Jun-ichi Tsujii, A Hybrid Japanese Parser with Hand-crafted Grammar and Statistics, in Proceedings of the 18th International Conference on Computational Linguistics, pp. 411-417, Saarbrucken, Germany, August, 2000.
- 吉永直樹, 宮尾祐介, 建石由佳, 鳥澤健太郎, 辻井潤一, FB-LTAG から HPSG への文法変換, 言語処理学会第 6 回年次大会発表論文集, pp. 183–186, March, 2000.
- 西田健二, 鳥澤健太郎, 辻井潤一, HPSG の複数の文脈自由文法へのコンパイル, 言語処理学会第 6 回年次大会発表論文集, pp. 187–190, March, 2000.
- 吉田稔, 鳥澤健太郎, 辻井潤一, 表形式からの情報抽出手法, 言語処理学会第 6 回年次大会発表論文集, pp. 252–255, March, 2000.
- 金山博, 鳥澤健太郎, 光石豊, 辻井潤一, 3 つ組・4 つ組による日本語係り受け解析, 言語処理学会第 6 回年次大会発表論文集, pp. 487–490, March, 2000.

- 宮尾祐介, 辻井潤一, 確率付き項構造による曖昧性解消, 言語処理学会第 6 回年次大会発表論文集, pp. 495–498, March, 2000.
- Sadao Kurohashi and Wataru Higasa: Automated Reference Service System at Kyoto University Library, In Proceedings of 2000 Kyoto International Conference on Digital Libraries: Research and Practice, pp.304-310, Kyoto (2000.11.13-16).
- Sadao Kurohashi and Manabu Ori: Nonlocal Language Modeling based on Context Co-occurrence Vectors, In Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp.80-86 (2000.10).
- Sadao Kurohashi and Wataru Higasa: Dialogue Helpsystem based on Flexible Matching of User Query with Natural Language Knowledge Base, In Proceedings of 1st ACL SIGdial Workshop on Discourse and Dialogue, pp.141-149, (2000.10).
- Daisuke Kawahara and Sadao Kurohashi: Japanese Case Structure Analysis by Unsupervised Construction of a Case Frame Dictionary, In Proceedings of 18th COLING, pp.432-438, (2000.8).
- Hideo Watanabe, Sadao Kurohashi and Eiji Aramaki: Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation, In Proceedings of 18th COLING, pp.906-912, (2000.8).
- Yasuhiko Watanabe, Yoshihiro Okada, Sadao Kurohashi and Eiichi Iwanari: Discourse Structure Analysis for News Video, In Processdings of IEEE International Conference on Multimedia and Expo (ICME2000), (2000.6).
- Richard F. E. Sutcliffe and Sadao Kurohashi: A Parallel English-Japanese Query Collection for the Evaluation of On-Line Help Systems, In Proceedings of The Second International Conference on Language Resources & Evaluation, pp.1665-1669, (2000.6).
- 村田真樹，山本専，黒橋禎夫，井佐原均，長尾真：名詞句「A の B」「AB」の用例を利用した換喩理解, 人工知能学会誌, Vol.15, No.3, pp.503-510 (2000.5).
- 白木伸征, 黒橋禎夫：自然言語入力と目次との柔軟な照合による図書検索システム, 情報処理学会論文誌, Vol.41, No.4, pp.1162-1170 (2000.4).

# 自然言語解析のための MSLR パーザ・ツールキット

白井　清昭[†]　植木　正裕[††]　橋本　泰一[†]
徳永　健伸[†]　田中　穂積[†]

本論文では，我々が現在公開している自然言語解析用ツール「MSLR パーザ・ツールキット」の特徴と機能について述べる．MSLR パーザは，一般化 LR 法の解析アルゴリズムを拡張し，日本語などの分かち書きされていない文の形態素解析と構文解析を同時に行うツールである．MSLR パーザを用いて解析を行う際には，まず LR 表作成器を用いて，文法と接続表から LR 表を作成する．このとき，LR 表作成器は，接続表に記述された品詞間の接続制約を組み込んだ LR 表を生成する．このため，接続制約に違反する解析結果を受理しない LR 表が作られるだけでなく，LR 表の大きさを大幅に縮小することができる．次に，MSLR パーザは，作成された LR 表と辞書を用いて辞書引きによる単語分割と構文解析を同時に行い，その結果として構文木を出力する．さらに，MSLR パーザは，文中の括弧の組によって係り受けに関する部分的な制約が与えられた文を入力とし，その制約を満たす構文木のみを出力する機能を持つ．また，文脈依存性を若干反映した言語モデルのひとつである確率一般化 LR モデル (PGLR モデル) を学習し，個々の構文木に対して PGLR モデルに基づく生成確率を計算し，解析結果の優先順位付けを行う機能も持つ．

**キーワード**:　形態素解析，構文解析，一般化 *LR* 法，パーザ

# MSLR Parser Tool Kit — Tools for Natural Language Analysis

Kiyoaki Shirai[†] , Masahiro Ueki[††] , Taiichi Hashimoto[†] ,
Takenobu Tokunaga[†] and Hozumi Tanaka[†]

In this paper, we describe a tool kit for natural language analysis, the MSLR parser tool kit. The 'MSLR parser' is based on the generalized LR parsing algorithm, and integrates morphological and syntactic analysis of unsegmented sentences. The 'LR table generator' constructs an LR table from a context free grammar and a connection matrix describing adjacency constraints between part-of-speech pairs. By incorporating connection matrix-based constraints into the LR table, it is possible to both reject any locally implausible parsing results, and reduce the size of the LR table. Then, using the generated LR table and a lexicon, the MSLR parser outputs parse trees based on morphological and syntactic analysis of input sentences. In addition to this, the MSLR parser accepts sentence inputs including partial syntactic constraints denoted by pairs of brackets, and suppresses the generation of any parse trees not satisfying those constraints. Furthermore, it can be trained according to the probabilistic generalized LR (PGLR) model, which is a mildly context sensitive language model. It can also rank parse trees in order of the overall probability returned by the trained PGLR model.

**KeyWords:**　*morphological analysis, syntactic analysis, generalized LR method, parser*

# 1　はじめに

我々は，1998 年 10 月から自然言語解析用ツール「MSLR パーザ・ツールキット」を公開している [1]．MSLR パーザ (Morphological and Syntactic LR parser) は，一般化 LR 法の解析アルゴリズムを拡張し，単語区切りのない言語 (日本語など) を主に対象とし，形態素解析と構文解析を同時に行うパーザである[2]．本論文では，MSLR パーザ・ツールキットの特徴と機能について述べる．

MSLR パーザを用いて文を解析する場合には，以下の 3 つが必要になる．

**文法**　品詞を終端記号とする文脈自由文法．主に構文解析に用いる．

**辞書**　単語とそれに対応した品詞を列挙したデータで，形態素解析の基本単位を集めたものである．辞書の品詞体系は文法の品詞体系と一致していなければならない．

**接続表**　品詞間の接続制約を記述した表．品詞間の接続制約とは，ある 2 つの品詞が隣接できるか否かに関する制約である．

本ツールキットでは，文法・辞書・接続表を自由に入れ換えることができる．すなわち，ユーザが独自に開発した文法や辞書を用いて，MSLR パーザによって文の解析を行うことが可能である．また，MSLR パーザ・ツールキットには日本語解析用の文法，辞書，接続表が含まれている．したがって，文法等を持っていないユーザでも，ツールキットに付属のものを用いて日本語文の形態素・構文解析を行うことができる．

MSLR パーザは C 言語で実装され，動作する OS は unix のみである．具体的には，以下の OS で動作することが確認されている．

- SunOS 5.6
- Digital Unix 4.0
- IRIX 6.5
- FreeBSD 3.3
- Linux 2.2.11, Linux PPC(PC-Mind 1.0.4)

MSLR パーザを動作させるために必要なメモリ使用量・ディスク使用量は，使用する文法や辞書の規模に大きく依存する．例えば，ツールキットに付属の日本語解析用文法 (規則数 1,408) と辞書 (登録単語数 241,113) を用いる場合，50Mbyte のメモリと 10Mbyte のディスク容量を必要

† 東京工業大学 大学院情報理工学研究科 計算工学専攻, Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology
†† 国立国語研究所 日本語教育センター 日本語教育普及指導部 日本語教育教材開発室, Teaching Materials Development Section, Department of Educational Support Services, Center for Teaching Japanese as a Second Language, The National Language Research Institute

1 `http://tanaka-www.cs.titech.ac.jp/pub/mslr/`

2 MSLR パーザは，分かち書きされた文 (英語文など) を解析する機能も持っているが，もともとは単語区切りのない文を解析することを目的に作られた．

とする．

本ツールキットを用いた形態素・構文解析の流れを図 1 に示す．MSLR パーザの解析アルゴリズムは一般化 LR 法に基づいているため，まず最初に LR 表作成器を用いて，文法と接続表から LR 表を作成する．MSLR パーザは，作成された LR 表と辞書を参照しながら入力文の形態素・構文解析を行い，解析結果 (構文木) を出力する．

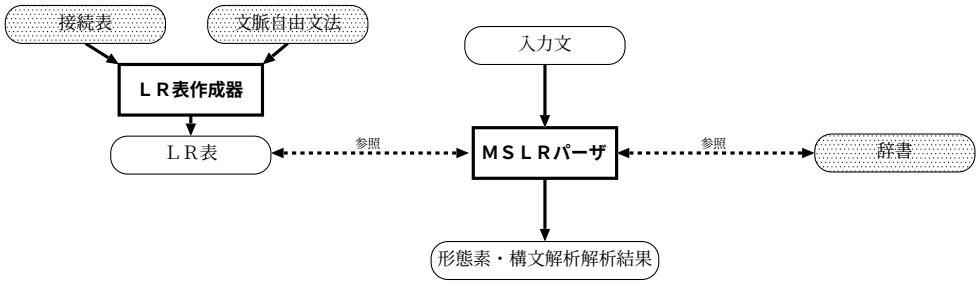



**図 1**　MSLR パーザを用いた形態素・構文解析の流れ

本ツールキットの主な特徴と機能は以下の通りである．

- MSLR パーザは，形態素解析と構文解析を同時に行う．まず最初に形態素解析を行い，その出力をもとに構文解析を行う逐次的な方法では，形態素解析の段階では文法などの構文的な制約を考慮しない場合が多く，その後の構文解析の段階で不適当と判断されるような無駄な解析結果も出力される．これに対し，MSLR パーザは形態的な情報 (辞書，接続表) と構文的な情報 (文法) を同時に用いて解析を行うため，このような無駄な解析結果を生成することはない．
- LR 表作成器は，接続表に記述された品詞間の接続制約を組み込んだ LR 表を作成する．すなわち，LR 表を作成する段階で品詞間の接続制約を考慮し，接続制約に違反する構文木を受理しない LR 表を作る．さらに，品詞間の接続制約を組み込んだ場合，接続制約を組み込まない場合と比べて LR 表の状態数・動作数を減らすことができ，メモリ使用量も小さくすることができるという利点がある．
- 品詞間の接続制約は，接続表という形式で記述する代わりに，文法に組み込むことも可能である．しかしながら，接続制約を文法に組み込んだ場合，規則数が組み合わせ的に増大する．このため，文法作成者の負担が大きくなり，また作成される LR 表の大きさも大きくなるために望ましくない．このような理由から，本ツールキットでは，接続表と文法を独立に記述する枠組を採用している．
- 平文を入力とした解析の他に，係り受けに関する部分的な制約を加えた文を入力とした解析を行うことができる．例えば,「太郎が渋谷で買った本を借りた」という文を解析す

る際に，以下のような括弧付けによる制約を付けた文が入力されたときには，括弧付けと矛盾した解析結果は出力しない．

[太郎が渋谷で買った] 本を借りた

すなわち,「太郎が」が「借りた」に係る以下のような解析結果は，A の括弧付けが入力の括弧付けと矛盾 (交差) しているために出力しない．

[[太郎が][$_A$ [[渋谷で][買った]][[本を][借りた]] $_A$]

この機能は，例えば前編集により係り受けに関する部分的な制約をあらかじめ文に付加してから解析を行い，構文的曖昧性を抑制する場合などに利用できる．

- 確率一般化 LR モデル (Inui, Sornlertlamvanich, Tanaka, and Tokunaga 1998; Sornlertlamvanich, Inui, Tanaka, Tokunaga, and Toshiyuki 1999) (Probabilistic Generalized LR Model，以下 PGLR モデル) を取り扱うことができる．PGLR モデルとは，一般化 LR 法の枠組において構文木の生成確率を与える確率モデルである．PGLR モデルに基づく構文木の生成確率は，統計的な意味での正しさの尺度を構文木に与えることができるので，構文的な曖昧性の解消に利用することができる．

以下では，ここに挙げた本ツールキットの特徴と機能について詳しく説明する．2 節では品詞間の接続制約を組み込む LR 表作成器について述べ，3 節では MSLR パーザの概略について述べる．最後に 4 節で本論文のまとめと MSLR パーザ・ツールキットの今後の開発方針について述べる．

## 2　LR 表作成器

本節では，MSLR パーザ・ツールキットにおける LR 表作成器の機能と特徴について詳しく説明する．

### 2.1　3 種類の LR 表を作成する機能

一般化 LR 法で用いられる LR 表には，SLR (Simple LR), CLR (Canonical LR), LALR (Lookahead LR) の 3 種類がある．我々の LR 表作成器は，これら 3 種類の LR 表を作成する機能を持つ．

実際の自然言語文の解析では，最も状態数の少ない LALR が用いられる場合が多い．したがって，以後 LR 表といえば LALR を意味するものとする．これらの LR 表の違いの詳細については文献 (Aho, Sethi, and Ullman 1985) を参照していただきたい．

## 2.2　品詞間の接続制約を組み込む機能

本ツールキットにおける LR 表作成器の最も大きな特徴は，LR 表に品詞間の接続制約を反映させることができる点にある．品詞間の接続制約を LR 表に反映させるということは，接続制約に違反する構文木を生成する動作を LR 表からあらかじめ除去することに相当する．

このことを図 2 の文法 $CFG_1$ を例に説明する[3]．$CFG_1$ において，書き換え規則の右側にある数字は規則番号を表わす．また，終端記号は品詞である．$CFG_1$ から通常の LR 表作成アルゴリズムによって作成された LR 表を図 3 に示す．但し，図 3 の LR 表は action 部のみであり，goto 部は省略されている．今，この LR 表に図 4 の接続表に記述された接続制約を反映させることを考える．図 4 の接続表において，行列要素 $(i, j)$ が 1 なら $i$ 行目の品詞 $x_i$ と $j$ 列目の品詞 $x_j$ がこの順序で連接可能であることを示し，$(i, j)$ が 0 なら $x_i$ と $x_j$ が連接不可能であることを意味する．また，“$” は文末を表わす特殊な品詞である．

| | | | |
|---|---|---|---|
| S → VP | (1) | VS1 → vs_1 | (10) |
| VP → PP VP | (2) | VS → vs_5k | (11) |
| PP → VP PP | (3) | VS → vs_5m | (12) |
| VP → V AX | (4) | VS → vs_5w | (13) |
| V → VS VE | (5) | VE → ve_i | (14) |
| V → VS1 | (6) | VE → ve_ki | (15) |
| PP → N P | (7) | VE → ve_ma | (16) |
| N → noun | (8) | AX → AX aux | (17) |
| P → postp | (9) | AX → aux | (18) |

**図 2**　文法の例: $CFG_1$

$CFG_1$ では，VS を構成する品詞として vs_5k, vs_5m, vs_5w が，VE を構成する品詞として ve_i, ve_ki, ve_ma があるので，規則 (5) から，V を構成する品詞列は $3 \times 3 = 9$ 通りあることがわかる．これに対し，図 4 の接続表を考慮した場合，これら 9 通りの品詞列のうち “vs_5k ve_ki”，“vs_5m ve_ma”，“vs_5w ve_i” の 3 組だけが接続制約を満たす．したがって，これら以外の品詞列は受理すべきではない．

ここで，図 3 の LR 表の状態 4，先読み記号 ve_i の欄にある re 11 という reduce 動作に着目する．re 11 は，$CFG_1$ における規則 (11) に対応した部分木を作ることを意味する (図 5)．と

3 $CFG_1$ における各記号のおおまかな意味は以下の通りである．S=文, VP=動詞句, PP=後置詞句, V=動詞, VS1=一段動詞語幹, VS=動詞語幹, VE=動詞語尾, N=名詞, P=助詞, AX=助動詞列 (以上，非終端記号)．vs_1=一段動詞語幹, vs_5k=カ行五段動詞語幹, vs_5m=マ行五段動詞語幹, vs_5w=ワ行五段動詞語幹, ve_i=動詞語尾イ, ve_ki=動詞語尾キ, ve_ma=動詞語尾マ, noun=名詞, postp=助詞, aux=助動詞 (以上，終端記号 (品詞))．

| | vs_1 | vs_5k | vs_5m | vs_5w | ve_i | ve_ki | ve_ma | aux | noun | postp | $ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | sh1 | sh4 | sh3 | sh2 | | | | | sh11 | | |
| 1 | | | | | | | | re10 | | | |
| 2 | | | | | re13 | *re13 | *re13 | | | | |
| 3 | | | | | *re12 | *re12 | re12 | | | | |
| 4 | | | | | *re11 | re11 | *re11 | | | | |
| 5 | | | | | | | | sh13 | | | |
| 6 | sh1 | sh4 | sh3 | sh2 | | | | | sh11 | | |
| 7 | | | | | | | | | | sh16 | |
| 8 | | | | | | | | re6 | | | |
| 9 | | | | | sh20 | sh19 | sh18 | | | | |
| 10 | sh1 | sh4 | sh3 | sh2 | | | | | sh11 | | re1 |
| 11 | | | | | | | | | | re8 | |
| 12 | | | | | | | | | | | acc |
| 13 | re18 | re18 | re18 | re18 | | | | re18 | re18 | | re18 |
| 14 | re4 | re4 | re4 | re4 | | | | sh24 | re4 | | re4 |
| 15 | re2/sh1 | re2/sh4 | re2/sh3 | re2/sh2 | | | | | re2/sh11 | | re2 |
| 16 | re9 | re9 | re9 | re9 | | | | | re9 | | |
| 17 | re7 | re7 | re7 | re7 | | | | | re7 | | |
| 18 | | | | | | | | re16 | | | |
| 19 | | | | | | | | re15 | | | |
| 20 | | | | | | | | re14 | | | |
| 21 | | | | | | | | re5 | | | |
| 22 | sh1 | sh4 | sh3 | sh2 | | | | | sh11 | | |
| 23 | re3/sh1 | re3/sh4 | re3/sh3 | re3/sh2 | | | | | re3/sh11 | | |
| 24 | re17 | re17 | re17 | re17 | | | | re17 | re17 | | re17 |

**図 3**　$CFG_1$ から生成される LR 表 (action 部のみ)

| | vs_1 | vs_5k | vs_5m | vs_5w | ve_i | ve_ki | ve_ma | noun | postp | aux | $ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| vs_1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| vs_5k | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| vs_5m | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| vs_5w | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| ve_i | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ve_ki | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ve_ma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| noun | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| postp | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| aux | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

**図 4**　接続表の例

ころが，先読み記号が ve_i であることから，“vs_5k ve_i” という品詞列に対してこの動作を実行することになるが，この品詞列は図 4 の接続制約に違反する．同様に，図 3 において，“*” のついた動作もまた接続制約に違反する動作である．したがって，このような動作を事前に LR 表から削除しておけば，接続制約に違反する解析結果の生成を防ぐことができる．

接続制約に違反する動作を LR 表から除去する方法としては，まず図 3 のように接続制約を考慮しない LR 表を作成してから，接続制約に違反する動作を LR 表から削除する方法が考え

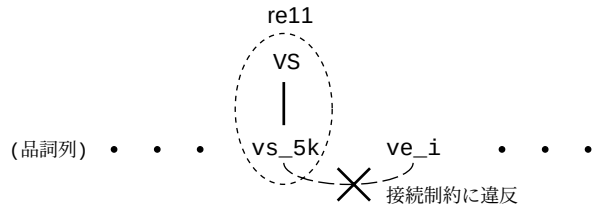



**図 5**　接続制約に違反する reduce 動作

$$\mathrm{V} \rightarrow \mathrm{VS\ VE} \quad (5) \qquad \Rightarrow \qquad \begin{array}{ll} \mathrm{V} \rightarrow \mathrm{vs\_5k\ ve\_ki} & (5\text{-}1) \\ \mathrm{V} \rightarrow \mathrm{vs\_5m\ ve\_ma} & (5\text{-}2) \\ \mathrm{V} \rightarrow \mathrm{vs\_5w\ ve\_i} & (5\text{-}3) \end{array}$$

**図 6**　接続制約を反映した文法規則

られる．しかしながら，文法の規模が大きくなると，接続制約を考慮しない LR 表の大きさが非常に大きくなるために望ましくない．これに対して，本ツールキットでは，LR 表を作成する段階で接続制約を考慮し，接続制約に違反する動作を除いた LR 表を直接生成する方法を採用している．接続制約を組み込みながら LR 表を作成するアルゴリズムの詳細については文献 (Li 1996) を参照していただきたい．

接続制約を LR 表に組み込む主な利点としては以下の 3 つが挙げられる．

(1)　接続制約を事前に組み込んだ LR 表を用いて解析を行った場合，解析時には品詞間の連接可能性をチェックする必要がないので，解析時の効率を上げることができる．

(2)　接続制約に違反する構文木を生成する動作を LR 表から除去することにより，LR 表の状態数・動作数を大幅に減らし，メモリ使用量を小さくすることができる．

(3)　品詞間の接続制約は，接続表として記述してから LR 表に組み込む代わりに，書き換え規則の細分化によって組み込むこともできる．例えば，$CFG_1$ の例では，規則 (5) の代わりに，図 6 に挙げる 3 つの規則を導入すれば，接続制約を満たす品詞列のみ受理することができる．しかしながら，このように接続制約を組み込んだ文法を作成することは，規則数が組み合わせ的に増大するために望ましくない．品詞間の接続制約は，接続表として文法とは独立に記述し，LR 表を作成する段階で接続制約を組み込む方が，最終的に得られる LR 表の状態数・動作数も少なく，メモリ使用量を小さくすることができる．また，文法記述者の負担も減らすことができる．

### 2.3 評価実験

LR 表に品詞間の接続制約を組み込む効果を調べる簡単な実験を行った．本ツールキットに付属されている日本語解析用の文法と接続表を用いて，品詞間の接続制約を組み込む場合と組み込まない場合の LR 表を比較した．使用した文法の規則数は 1,408，非終端記号数は 218，終端記号数は 537 である．実験に使用した計算機は Sun Ultra Enterprise 250 Server(主記憶 2GB, CPU 周波数 300MHz) である．結果を表 1 に示す．

**表 1**　品詞間の接続制約を LR 表に組み込むことの効果

| | CPU 時間 | 状態数 | 動作数 |
|---|---|---|---|
| 接続制約なし | 42.1(sec.) | 1,720 | 379,173 |
| 接続制約あり | 45.4(sec.) | 1,670 | 197,337 |

表 1 において,「CPU 時間」は LR 表作成に要した CPU 時間を,「状態数」は作成された LR 表の状態の数を,「動作数」は作成された LR 表の動作 (shift 動作と reduce 動作) の数を示している．この表から，品詞間の接続制約を組み込むことによって，状態数はほとんど変わらないが，動作数は約半分に減ることがわかる．したがって，LR 表のために必要なメモリ使用量を大幅に縮小することができる．一方,「CPU 時間」は，接続制約を考慮する場合としない場合とでそれほど大きな差は見られなかった．一般に，接続制約を組み込む場合は，品詞間の連接可能性を調べながら LR 表を作成するために，それに要する時間は長くなることが予想される．しかしながら，接続制約に違反する無駄なアイテムが生成されなくなることから，LR 表作成に要する時間が短縮される効果も生じる．そのため，LR 表作成時間が劇的に増大するわけではないことが実験的に確かめられた．

## 3 MSLR パーザ

本節では，MSLR パーザの機能と特徴について概説する．

### 3.1 形態素解析と構文解析を同時に行う機能

1 節で述べたように，MSLR パーザは形態素解析と構文解析を同時に行う (Tanaka, Tokunaga, and Aizawa 1995)．また，形態素・構文解析結果として構文木を出力する．例えば，図 2 の文法 ($CFG_1$)，図 4 の接続表，図 7 の辞書を用いたときの「あいこにたのまれた」という文の解析結果 (構文木) を図 8 に示す．実際には，MSLR パーザは以下のような括弧付けで表現された構文木を出力する．

| 単語 | 品詞 | 単語 | 品詞 |
|---|---|---|---|
| あ | vs_5k, vs_5w | たの | vs_5m |
| あいこ | noun | に | postp, vs_1 |
| い | ve_i | の | vs_5m |
| き | ve_ki | ま | ve_ma |
| た | aux | れ | aux |

**図 7**　辞書の例

```
[<S>,[<VP>,[<PP>,[<N>,[noun, あいこ]],[<P>,[postp, に]]],[<VP>,[<V>,[<VS>,
[vs_5m, たの]],[<VE>,[ve_ma, ま]]],[<AX>,[<AX>,[aux, れ]],[aux, た]]]]]
```

解析結果が複数ある場合には，その中から $N$ 個の構文木をランダムに選んで出力する．ただし，3.3 項で述べる PGLR モデルを用いる場合には，構文木の生成確率の大きい上位 $N$ 個の構文木を取り出すことができる．また，$N$ の値は起動時のオプション指定により変更できる．

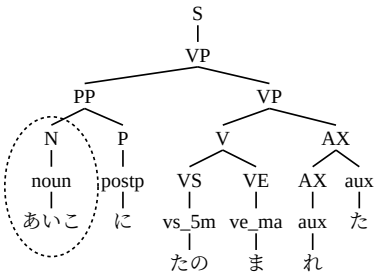



**図 8**　「あいこにたのまれた」の解析結果

MSLR パーザのアルゴリズムは，一般化 LR 法の構文解析アルゴリズムを拡張したものである．一般化 LR 法が通常は品詞列を入力とするのに対して，MSLR パーザは文字列を入力とし，辞書引きによる単語分割と構文解析を同時に行う．以下，一般化 LR 法と MSLR パーザの解析アルゴリズムとの違いを簡単に説明する．MSLR パーザの解析アルゴリズムの詳細については文献 (Tanaka et al. 1995) を参照していただきたい．

(1)　入力文が与えられたとき，品詞と品詞の間に位置番号をつける代わりに，図 9 のように入力文の文字間に位置番号をつける．

(2)　解析が位置 $i$ まで進んだとき，位置 $i$ から始まる全ての単語を辞書引きし，その結果をスタックに登録する．例えば，図 9 の例文を図 7 の辞書を用いて解析した場合，位置 0 では “(あ,vs_5k)”,“(あ,vs_5w)”,“(あいこ,noun)” という 3 つの品詞付けの結果が解析スタックに登録される．これらの品詞付けの結果は，通常の一般化 LR 法におけ

あ い こ に た の ま れ た
(位置番号)　0　1　2　3　4　5　6　7　8　9

**図 9**　MSLR パーザにおける位置番号のつけ方

る多品詞語と全く同様に取り扱われる．

(3)　shift 動作を実行して先読み記号をスタックにプッシュする際には，その品詞を構成する文字列の一番最後の位置まで解析スタックを延ばす．例えば，位置 0 で vs_5k という先読み記号 (品詞) をプッシュする際には，vs_5k が位置 0〜1 に位置する単語「あ」の品詞であるので，スタックの先頭を位置 1 まで延ばす．そして，位置 1 から始まる単語の辞書引き結果をもとに以後の解析をすすめる．同様に，位置 0 で noun という品詞をプッシュする際には，noun が位置 0〜3 に位置する単語「あいこ」の品詞であるので，スタックの先頭を位置 3 まで延ばす．以後の解析は，位置 3 から始まる単語の辞書引き結果をもとに進められる．

例文「あいこにたのまれた」を解析する際，形態素解析結果の候補としては以下の 2 つがある．

a.　(あいこ,noun)(に,postp)(たの,vs_5m)(ま,ve_ma)(れ,aux)(た,aux)

b.　(あいこ,noun)(に,vs_1)(た,aux)(の,vs_5m)(ま,ve_ma)(れ,aux)(た,aux)

文法 $CFG_1$ は b. の品詞列を受理しないが，形態素解析と構文解析を逐次的に行う方法では，形態素解析結果の候補として a., b. ともに出力し，それぞれの品詞列に対して構文解析が試みられる．これに対し，MSLR パーザは形態素解析と構文解析を同時に行い，文法に記述された構文的な制約で排除される形態素解析の結果を早期に取り除くことができるため，解析効率がよい．例えば，位置 3 まで解析が進んだとき,「あいこ」という文字列が図 8 の点線で囲まれた部分木を構成することがわかっている．このとき，位置 3 から始まる単語を辞書引きする際に，品詞列 b. は受理されないという文法的な制約から，“(に,vs_1)” という品詞付けが適切でないことがわかる．具体的には，位置 3 におけるスタックトップの状態 7 において，“vs_1” を先読み記号とする動作が図 3 の LR 表に存在しないことから，“(に,vs_1)” という辞書引き結果を含む解析はこの時点で中断される．したがって，誤りである形態素解析結果の候補 b. を早期に取り除くことができる．このことは，MSLR パーザの大きな特徴の 1 つである．

## 3.2　括弧付けによる制約のついた入力文を解析する機能

MSLR パーザは括弧付けによる制約を加えた文を解析することができる．具体的には，MSLR パーザは以下のような文字列を入力として，括弧付けに矛盾しない解析結果のみを出力する機能を持つ．

[*, 太郎が渋谷で買った] 本を借りた

この例では括弧による制約はひとつしかないが，括弧による制約は複数あってもよい．また，複数の制約が入れ子になっても構わない．以下に例を挙げる．

[*, 太郎が [*, 渋谷で買った]] [*, 本を借りた]

上記の入力例において，“*” は括弧で示された範囲を支配する非終端記号に特に制約がないことを表わしている．これに対し，“*” の位置に非終端記号を指定することにより，括弧に矛盾する解析結果だけでなく，括弧で囲まれた文字列を支配する非終端記号を限定することもできる．例えば，以下のような入力に対して，MSLR パーザは「あいこに」を支配する非終端記号が “<PP>” となる解析結果のみを出力する．

[<PP>, あいこに] たのまれた

括弧付けによる制約を取り扱う機能は，前編集によりあらかじめ部分的な制約を付加する際に利用することができる．構文解析を完全に自動で行うのではなく，インタラクティブに人間の知識を利用しながら半自動的に構文解析を行うことは，解析精度を向上させる有効な手段のひとつである．解析を行う前に，係り受けに関する部分的な制約をうまく人手で与えれば，構文的曖昧性を激的に減らすことができ，結果として構文解析の精度を飛躍的に向上させることが期待できる．

## 3.3 PGLR モデルを取り扱う機能

PGLR モデル (Inui et al. 1998) は，一般化 LR 法の枠組に基づいて構文木の生成確率を与える確率モデルである．PGLR モデルにおける構文木の生成確率は，構文木を作り出す際に実行される LR 表上の動作 (shift 動作もしくは reduce 動作) の実行確率の積として推定される．この生成確率は，生成される複数の構文木の中から最も正しい構文木を選択する構文的曖昧性解消に利用できる．ここで注意すべき点は，PGLR モデルによって与えられる構文木の生成確率は品詞を葉とする構文木の生成確率だということである．すなわち，単語の導出確率や単語の共起関係などの語彙的な統計情報は考慮されていない[4]．

LR 表の動作の実行確率には若干の文脈依存性が反映されていると考えられる．したがって，PGLR モデルは，文脈自由な言語モデルである確率文脈自由文法よりも推定パラメタ数は多くなるが，文脈依存性が考慮されたより精密なモデルを学習することが可能であり，構文的曖昧性解消の精度も向上することが実験的にも確かめられている (Sornlertlamvanich et al. 1999)．

本ツールキットでは，PGLR モデルを学習する機能，及び PGLR モデルによる構文木の生成確率を計算する機能を備えている．以下，それぞれの機能の概要について説明する．

4 PGLR モデルと，PGLR モデルとは独立に学習された語彙的な統計情報を組み合わせて構文解析を行う試みも行われている (白井, 乾, 徳永, 田中 1998).

### 3.3.1 PGLR モデルの学習について

PGLR モデルの学習は，LR 表上の各動作の実行確率を推定することにより行われる．動作の実行確率の推定に必要なものは，構文木が付与された構文木付きコーパスである．まず，例文に付与された構文木に対して，構文木を生成する際に実行する LR 表上の動作の使用回数 $C(s_i, l_j, a_k)$ を数え上げる．ここで，$s_i$ は LR 表における状態を，$l_j$ は先読み記号を，$a_k$ は動作を表わし，$C(s_i, l_j, a_k)$ は，状態が $s_i$ で先読み記号が $l_j$ のときに動作 $a_k$ が実行された回数を表わす．

LR 表上の各動作の実行確率は式 (1)(2) によって推定する．

$$P(l_j, a_k|s_i) = \frac{C(s_i, l_j, a_k)}{\sum_{j,k} C(s_i, l_j, a_k)} \quad if \quad s_i \in S_s \tag{1}$$

$$P(a_k|s_i, l_j) = \frac{C(s_i, l_j, a_k)}{\sum_{k} C(s_i, l_j, a_k)} \quad if \quad s_i \in S_r \tag{2}$$

式 (1)(2) において，$S_s$ は shift 動作直後に到達する状態の集合，$S_r$ はそれ以外の状態の集合を表わす．LR 表における全ての状態は $S_s$ または $S_r$ のどちらか一方に必ず属する．図 3 の LR 表の例では，$S_s = \{0, 1, 2, 3, 4, 11, 13, 16, 18, 19, 20, 24\}$, $S_r = \{5, 6, 7, 8, 9, 10, 12, 14, 15, 17, 21, 22, 23\}$ である．初期状態 0 は $S_s$ に属することに注意していただきたい．

式 (1) は，$s_i \in S_s$ のときには，状態 $s_i$ で実行されうる全ての動作で実行確率を正規化することを意味する．言い換えれば，LR 表における同じ行に属する動作の実行確率の和は 1 となる．例えば，図 3 の LR 表の状態 0 にある 5 つの shift 動作は，これらの実行確率の和が 1 になるように正規化される．これに対して式 (2) は，$s_i \in S_r$ のときには，状態 $s_i$，先読み記号 $l_i$ のときに実行されうる全ての動作で実行確率を正規化することを意味する．すなわち，LR 表における同じマス目に属する動作の実行確率の和は 1 となる．例えば，図 3 の LR 表の状態 15，先読み記号 vs_1 の欄にある 2 つの動作 (re 2 と sh 1) の実行確率は，これらの和が 1 になるように正規化される．また，$S_r$ に属する状態の場合，shift/reduce コンフリクトがない限り，その状態に属する動作の実行確率は必ず 1 となる．

本ツールキットにおける PGLR モデル学習の手続きは以下の通りである．まず，MSLR パーザは，構文解析を行う際に，LR 表の各動作の使用回数を出力する機能を持っている．さらに，3.2 項で述べた括弧付けによる制約を取り扱う機能を利用し，訓練用コーパスに付与された構文木を入力として解析を行うことにより，訓練用コーパス中の構文木を生成する際に使われた各動作の使用回数 $C(s_i, l_j, a_k)$ を求めることができる．また本ツールキットには，このようにして得られた $C(s_i, l_j, a_k)$ から式 (1)(2) に従って各動作の実行確率を推定し，その実行確率が付与された LR 表を作成するツールが含まれている．このツールは，パラメタ推定の平滑化のために，LR 表に登録されている全ての動作の実行回数にある一定の頻度を加える機能を備えている．

**表 2**　解析実験の結果

| | Set A | Set B |
|---|---|---|
| 平均単語数 | 8.12 | 19.6 |
| 平均解析木数 | 13.1 | 15,500 |
| 平均解析時間 (ms) | 6.53 | 27.7 |

### 3.3.2　PGLR モデルを用いた解析について

MSLR パーザは，解析結果となる構文木とその PGLR モデルに基づく生成確率を同時に出力することができる．また，生成確率の高い順に構文木を並べて出力することができる．すなわち，PGLR モデルに基づく生成確率を用いた解析結果の優先順位付けを行うことができる．

MSLR パーザは，まず文法が受理する全ての解析結果を求め，それらをまとめた圧縮統語森を生成する．次に，この圧縮統語森を展開して個々の構文木を出力する際に，PGLR モデルに基づく構文木の生成確率を考慮し，生成確率の上位の構文木から優先して出力する．解析の途中で生成確率の低い部分木を除去するなどの枝刈りを行っていないため，生成確率の上位 $N$ 位の構文木が必ず得られることが保証される代わりに，長文など構文的曖昧性が非常に多い文を解析する際にメモリ不足によって解析に失敗する可能性も高い．したがって，我々は解析途中で生成確率の低い部分木を除去して探索空間を絞り込む機構も必要であると考えている．Sornlertlamvanich は PGLR モデルを利用した効率の良い枝刈りのアルゴリズムを提案しているが (Sornlertlamvanich 1998)，現在公開している MSLR パーザには実装されていない．

## 3.4　解析例

本項では，MSLR パーザを用いた簡単な日本語文解析実験について報告する．実験用コーパスとして，ATR が作成した日本語対話コーパス (Morimoto, Uratani, Takezawa, Furuse, Sobashima, Iida, Nakamura, and Sagisaka 1994) を使用した．実験に用いた文法は，対話文解析用の文脈自由文法で，非終端記号数 172，終端記号数 441，規則数は 860 である (田中, 竹澤, 衛藤 1997)．今回の実験では，日本語対話コーパス約 20,000 文のうち，上記の文法による構文木が付与された例文 10,020 文を使用した．辞書及び接続表は，これら 10,020 文から自動的に作成した．

評価用テキストとして，単語数 4〜14，15 以上の文をランダムに 1000 文ずつ取り出し，それぞれ Set A, Set B とした．これらの評価用例文について，分かち書きされていない文字列を入力とし，MSLR パーザを用いて形態素・構文解析を行った．また，評価用テキスト以外の例文約 9000 文から PGLR モデルを学習し，その PGLR モデルに基づく構文木の生成確率によって解析結果の順位付けを行った．使用した計算機は，2.3 項の実験と同じ Sun Ultra Enterprise 250 Server である．実験結果を表 2, 3 に示す．また，解析結果の具体例を付録 A に示す．

**表 3**　解析実験の結果 (文正解率)

| | 【形態素解析の文正解率】 | | 【構文解析の文正解率】 | |
|---|---|---|---|---|
| $n$ | Set A | Set B | Set A | Set B |
| 1 | 88.3% | 63.7% | 80.1% | 36.3% |
| 2 | 94.4% | 75.1% | 90.6% | 50.4% |
| 3 | 96.8% | 80.6% | 95.0% | 58.8% |
| 4 | 97.6% | 83.6% | 96.4% | 65.0% |
| 5 | 98.8% | 87.2% | 97.6% | 69.6% |

表 2 において,「平均解析木数」は 1 文あたりに生成される構文木の平均であり,「平均解析時間」は 1 文の解析に要した時間 (単位はミリ秒) の平均を表わしている．Set A のような短い文の場合は 7 ミリ秒程度，Set B のような長めの文の場合でも 27 ミリ秒程度で解析を行うことができる．また，表 3 の【形態素解析の文正解率】は，PGLR モデルに基づく構文木の生成確率の上位 $n$ 位の解析結果の中に，単語分割と品詞付けの結果がコーパスに付加されたものと一致する構文木が含まれる文の割合を表わしている．同様に【構文解析の文正解率】は，上位 $n$ 位の解析結果の中にコーパスに付加されたものと一致する構文木が含まれる文の割合を示している．この表から，例えば生成確率の 1 位の構文木について，Set A では約 80%，Set B では約 36%の文に対して正しい形態素・構文解析結果が得られたことがわかる．今回の実験で使用したコーパスがドメインの限られたコーパスであり，また辞書と接続表を評価用テキストと訓練用テキストの両方を用いて作成したこともあり，比較的良い結果が得られている．

## 4　おわりに

本論文では，我々が現在公開している自然言語解析用ツール「MSLR パーザ・ツールキット」の機能と特徴について述べた．最後に，本ツールキットの今後の開発方針について述べる．

まず，複数の接続制約を同時に組み込む LR 表作成器，さらにそれを用いて解析を行うパーザの実装を進めている．現在のツールでは，LR 表に組み込める接続制約の数は 1 種類のみである．しかしながら，例えば音声認識と同時に構文解析を行う場合，品詞間の接続制約だけでなく，音素間の接続制約も同時に利用した方が効率の良い解析ができると考えられる (今井 1999)．この場合，音素と品詞の 2 つの接続制約を LR 表に組み込む必要がある．また，これに合わせて，MSLR パーザの解析アルゴリズムも変更する必要がある．現在，複数の制約を取り扱う LR 表作成器および MSLR パーザのプロトタイプは完成しているが，効率の面でまだ問題があり，改良を進めている．

次に，よりロバストな解析ができるようにパーザを拡張することが挙げられる．特に，辞書にない単語 (未知語) が入力文中に現われたときには，原則的には解析に失敗する．現在の

MSLR パーザは，カタカナが続いた文字列を未知語として登録するなど，非常に簡単な未知語処理機能が付加されているが，まだ改良の余地も多い．また，解析に失敗した場合でも，部分的な解析結果を表示する機能なども追加していきたいと考えている．

最後に，本ツールキットに付属の日本語解析用の文法，辞書，接続表を改良することが今後の課題として挙げられる．これらを用いて新聞記事の解析を行った場合，解析に成功して何らかの結果を返すことのできる文の割合は約 85%である．解析に失敗する原因としては，前述の未知語処理の不完全さや文法規則の不備によるものが多い．より多様な文を解析できるようにするためには，特に文法を改良していかなければならない．また，本ツールキットに付属の文法を用いて解析を行った場合，PGLR モデルを学習するための構文木付きコーパスが存在しないために，PGLR モデルに基づく生成確率によって解析結果に優先順位を付けることはできない[5]．現在，構文木付きコーパスを必要としない PGLR モデルの学習方法について研究をすすめている．

**謝辞**

MSLR パーザ・ツールキットは多くの方の協力を得て開発されました．李輝氏，日本アイ・ビー・エム株式会社の綾部寿樹氏には初期の LR 表作成器を実装していただきました．九州工業大学の乾健太郎助教授には，PGLR モデルの理論及び実装について議論していただきました．Sussex 大学の John Carroll 氏，National Electronics and Computer Technology Center の Sornlertlamvanich Virach 氏には，MSLR パーザの実装に関する貴重な助言をいただきました．以上の皆様を始め，本ツールキットの開発に御協力いただきました全ての人々に感謝いたします．

MSLR パーザの辞書引きモジュールは，奈良先端科学技術大学院大学・松本研究室で開発された高速文字列検索システム SUFARY をベースに作成しています．SUFARY の転用を許可下さいました松本研究室の皆様に深く感謝いたします．

本ツールキットに付属の日本語解析用の辞書は，日本電子化辞書研究所が作成した EDR 日本語単語辞書 (日本電子化辞書研究所 1995) をもとに構築されています．本辞書の公開を許可下さいました日本電子化辞書研究所の皆様に深く感謝いたします．

# 参考文献

Aho, A. V., Sethi, R., and Ullman, J. D. (1985). *Compilers — principles, techniques, and tools*. Addison Wesley.

今井宏樹 (1999). 音声認識のための PGLR パーザに関する研究. Ph.D. thesis, Department of Computer Science, Tokyo Institute of Technology. `ftp://ftp.cs.titech.ac.jp/pub/TR/99/TR99-0016.ps.gz`.

5 公開されているツールでは，付属の文法を用いて解析を行った場合でも，単語数最小法，文節数最小法のヒューリスティクスに基づく解析結果の優先順位付けを行うことができる．

Inui, K., Sornlertlamvanich, V., Tanaka, H., and Tokunaga, T. (1998). "Probabilistic GLR Parsing: A new Formalization and Its Impact on Parsing Performance." 自然言語処理, **5** (3), 33–52.

Li, H. (1996). *Integrating Connection Constraints into a GLR Parser and its Applications in a Continuous Speech Recognition System.* Ph.D. thesis, Department of Computer Science, Tokyo Institute of Technology. `ftp://ftp.cs.titech.ac.jp/pub/TR/96/TR96-0003.ps.gz`.

Morimoto, T., Uratani, N., Takezawa, T., Furuse, O., Sobashima, O., Iida, H., Nakamura, A., and Sagisaka, Y. (1994). "A Speech and Language Database for Speech Translation Research." In *Proceedings of the International Conferrence on Spoeken Language Processing*, pp. 1791–1794.

日本電子化辞書研究所 (1995). "EDR 電子化辞書仕様説明書第 2 版." テクニカル・レポート TR–045.

白井清昭, 乾健太郎, 徳永健伸, 田中穂積 (1998). "統計的構文解析における構文的統計情報と語彙的統計情報の統合について." 自然言語処理, **5** (3), 85–106.

Sornlertlamvanich, V. (1998). *Probabilistic Language Modeling for Generalized LR Parsing.* Ph.D. thesis, Department of Computer Science, Tokyo Institute of Technology. `ftp://ftp.cs.titech.ac.jp/pub/TR/98/TR98-0005.ps.gz`.

Sornlertlamvanich, V., Inui, K., Tanaka, H., Tokunaga, T., and Toshiyuki, T. (1999). "Empirical Support for New Probabilistic Generalized LR Parsing." 自然言語処理, **6** (3), 3–22.

田中穂積, 竹澤寿幸, 衛藤純司 (1997). "MSLR 法を考慮した音声認識用日本語文法–LR 表工学 (3)–." 情報処理学会音声言語情報処理研究会, 97 巻, pp. 145–150.

Tanaka, H., Tokunaga, T., and Aizawa, M. (1995). "Integration of Morphological and Syntactic Analysis based on LR Parsing Algorithm." 自然言語処理, **2** (2), 59–73.

## 略歴

**白井 清昭:** 1993 年東京工業大学工学部情報工学科卒業．1995 年同大学院理工学研究科修士課程修了．1998 年同大学院情報理工学研究科博士課程修了．同年同大学院情報理工学研究科計算工学専攻助手，現在に至る．博士 (工学)．統計的自然言語解析に関する研究に従事．情報処理学会会員．

**植木 正裕:** 1995 年東京工業大学工学部情報工学科卒業．1997 年同大学院情報理工学研究科修士課程修了．2000 年同大学院情報理工学研究科博士課程満期退学．同年 4 月同大学院情報理工学研究科計算工学専攻技術補佐員．同年 7 月国立国語研究所日本語教育センター研究員，現在に至る．自然言語解析に

関する研究に従事．情報処理学会会員．

**橋本 泰一:** 1997 年東京工業大学工学部情報工学科卒業．1999 年同大学院情報理工学研究科計算工学専攻修士課程修了．同年同大学院情報理工学研究科計算工学専攻博士課程進学，在学中．統計的自然言語解析に関する研究に従事．

**徳永 健伸:** 1983 年東京工業大学工学部情報工学科卒業．1985 年同大学院理工学研究科修士課程修了．同年 (株) 三菱総合研究所入社．1986 年東京工業大学大学院博士課程入学．現在, 同大学大学院情報理工学研究科計算工学専攻助教授．博士 (工学)．自然言語処理, 計算言語学に関する研究に従事．情報処理学会, 認知科学会, 人工知能学会, 計量国語学会, Association for Computational Linguistics, 各会員．

**田中 穂積:** 1964 年東京工業大学工学部情報工学科卒業．1966 年同大学院理工学研究科修士課程修了．同年電気試験所 (現電子技術総合研究所) 入所．1980 年東京工業大学助教授．1983 年東京工業大学教授．現在, 同大学大学院情報理工学研究科計算工学専攻教授．博士 (工学)．人工知能，自然言語処理に関する研究に従事．情報処理学会, 電子情報通信学会，認知科学会, 人工知能学会, 計量国語学会, Association for Computational Linguistics, 各会員．

( 年　月　日 受付)
( 年　月　日 採録)

# 付録

# A　MSLR パーザによる解析例

3.4 項の実験で得られた解析結果の例を挙げる．まず，以下の例文 (1),(2),(3) を解析し，PGLR モデルによる生成確率の最も大きい解析結果のみを表示させたときの MSLR パーザの出力を示す.

(1)　七日までのご予約ですので八日と九日の分でございますか
(2)　十日と十一日のご予約を十一日と十二日に変更なさりたいわけですね
(3)　御社の場合には割引価格が適用されますので朝食も含めて割と良いお部屋を百九十三ドルでご提供できます

**• MSLR パーザの出力**

```
% mslr -g atr.gra -l atr.prtb.set2 -d atr-all.dic.ary -i -p -P -N 1 < sentence
reading the grammar file 'atr.gra' Done
reading LR table file 'atr.prtb.set2' Done
### 1 ###
$TAC23034-0030-3
七日までのご予約ですので八日と九日の分でございますか

accept
```

[<sent>,[<cl>,[<adv-cl>,[<verb>,[<verb/ga>,[<np>,[<n-sahen>,[<mod-n>,[<pp>,[<np>,[<n-date&time>,[<n-day>,[meisi-hi, 七日]]]],[<p-kaku-optn>,[p-kaku-made, まで]]],[<p-rentai>,[p-rentai-no, の]]],[<n-sahen>,[<n-sahen/ga-o>,[<prefix>,[prefix-go, ご]],[sahen-meisi/ga-o, 予約]]]]],[<aux>,[<auxstem>,[auxstem-desu, で]],[<infl>,[infl-spe-su, す]]]]],[<p-conj-advcl>,[p-conj-syusi, ので]]],[<cl>,[<vaux>,[<vaux>,[<verb>,[<verb/ga>,[<np>,[<n-hutu>,[<mod-n>,[<np>,[<n-date&time>,[<mod-n>,[<np>,[<n-date&time>,[<n-day>,[meisi-hi, 八日]]]],[<p-para>,[p-para-to, と]]],[<n-date&time>,[<n-day>,[meisi-hi, 九日]]]]],[<p-rentai>,[p-rentai-no, の]]],[<n-hutu>,[hutu-meisi-post, 分]]]],[<aux>,[aux-de, で]]]],[<aux>,[<auxstem>,[auxstem-copula-masu, ございま]],[<infl>,[infl-spe-su, す]]]],[<aux>,[aux-sfp-ka, か]]]]]] 5.716416e-23

total 1314
CPU time 0.2 sec

### 2 ###
$TAS13004-0100-1
十日と十一日のご予約を十一日と十二日に変更なさりたいわけですね

accept

[<sent>,[<cl>,[<vaux>,[<verb>,[<verb/ga>,[<np>,[<vaux>,[<vaux>,[<verb>,[<verb/ga>,[<pp-o>,[<np>,[<n-sahen>,[<mod-n>,[<np>,[<n-date&time>,[<mod-n>,[<np>,[<n-date&time>,[<n-day>,[meisi-hi, 十日]]]],[<p-para>,[p-para-to, と]]],[<n-date&time>,[<n-day>,[meisi-hi, 十一日]]]]],[<p-rentai>,[p-rentai-no, の]]],[<n-sahen>,[<n-sahen/ga-o>,[<prefix>,[prefix-go, ご]],[sahen-meisi/ga-o, 予約]]]]],[p-kaku-o, を]],[<verb/ga-o>,[<mod-v>,[<pp>,[<np>,[<n-date&time>,[<mod-n>,[<np>,[<n-date&time>,[<n-day>,[meisi-hi, 十一日]]]],[<p-para>,[p-para-to, と]]],[<n-date&time>,[<n-day>,[meisi-hi, 十二日]]]]],[<p-kaku-optn>,[p-kaku-ni, に]]]],[<n-sahen/ga-o>,[sahen-meisi/ga-o, 変更]]]]],[<aux>,[<auxstem>,[auxstem-sahen-5-r, なさ]],[<infl>,[infl-5-ri, り]]]],[<aux>,[<auxstem>,[auxstem-wish, た]],[<infl>,[infl-adj-i, い]]]],[<np>,[<n-hutu>,[n-keisiki, わけ]]]],[<aux>,[<auxstem>,[auxstem-desu, で]],[<infl>,[infl-spe-su, す]]]]],[<aux>,[aux-sfp-ne, ね]]]]] 6.846102e-32

total 2583
CPU time 0.3 sec

### 3 ###
$TAS12006-0080-1
御社の場合には割引価格が適用されますので朝食も含めて割と良いお部屋を百九十三ドルでご提供できます

accept

[<sent>,[<cl>,[<adv-cl>,[<vaux>,[<vaux>,[<vaux>,[<verb>,[<verb/o>,[<mod-v>,[<pp>,[<pp>,[<np>,[<n-hutu>,[<mod-n>,[<np>,[<n-hutu>,[hutu-meisi, 御社]]],[<p-rentai>,[p-rentai-no, の]]],[<n-hutu>,[hutu-meisi, 場合]]]],[<p-kaku-optn>,[p-kaku-ni, に]]],[<p-kakari>,[p-kakari-wa, は]]]],[<verb/o>,[<pp-ga>,[<np>,[<n-hutu>,[<n-sahen>,[<n-sahen/ga-o>,[sahen-meisi/ga-o, 割引]]],[<n-hutu>,[hutu-meisi, 価格]]]],[p-kaku-ga, が]],[<n-sahen/ga-o>,[sahen-meisi/ga-o, 適用]]]],[<aux>,[aux-suru-sa, さ]]],[<aux>,[<auxstem-deac>,[auxstem-deac-reru, れ]]]],[<aux>,[<auxstem>,[auxstem-masu, ま]],[<infl>,[infl-spe-su, す]]]],[<p-conj-advcl>,[p-conj-syusi, ので]]],[<cl>,[<adv-cl>,[<verb>,[<verb/ga-ni-o>,[<mod-v>,[<pp>,[<np>,[<n-hutu>,[hutu-meisi, 朝食]]],[<p-kakari>,[p-kakari-mo, も]]]],[<verb/ga-ni-o>,[<vstem1/ga-ni-o>,[vstem-1/ga-ni-o, 含め]]]]],[<p-conj-advcl>,[p-conj-renyo-te, て]]],[<cl>,[<vaux>,[<vaux>,[<verb>,[<verb/ga>,[<pp-o>,[<np>,[<n-hutu>,[<mod-n>,[<verb>,[<verb/ga>,[<mod-v>,[<advp>,[<adv>,[hukusi, 割と]]]],[<verb/ga>,[adjstem/ga, 良],[<infl>,[infl-adj-i, い]]]]],[<n-hutu>,[<prefix>,[prefix-o, お]],[<n-hutu>,[hutu-meisi, 部屋]]]]],[p-kaku-o, を]],[<verb/ga-o>,[<mod-v>,[<pp>,[<np>,[<n-quant>,[<n-num>,[<n-num-hyaku>,[<n-num-keta-hyaku>,[<num-suf-hyaku>,[num-hyaku, 百]]],[<n-num-zyuu>,[<n-num-keta-zyuu>,[<n-num-ichi>,[num-kyuu, 九]],[<num-suf-zyuu>,[num-zyuu, 十]]],[<n-num-ichi>,[num-san, 三]]]]],[<suffix-unit>,[suffix-doru, ドル]]]],[<p-kaku-optn>,[p-kaku-de, で]]]],[<n-sahen/ga-o>,[<prefix>,[prefix-go, ご]],[sahen-meisi/ga-o, 提供]]]]],[<aux>,[<auxstem>,[auxstem-sahen-1, でき]]]],[<aux>,[<auxstem>,[auxstem-masu, ま]],[<infl>,[infl-spe-su, す]]]]]]]] 6.264841e-45

total 19284
CPU time 0.13 sec

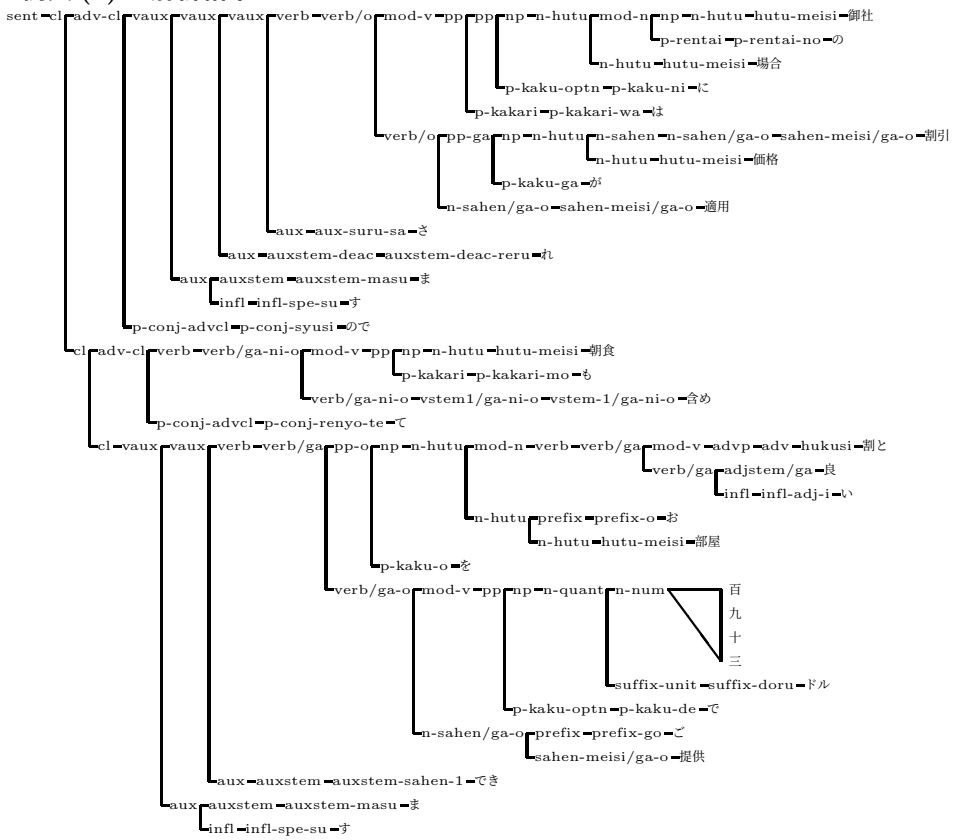
解析結果は括弧付けで表現された構文木として出力される．構文木の右にある数値はその構文木の PGLR モデルによる生成確率である．「total」は得られた解析結果の総数を，「CPU time」は解析に要した時間を表わす．

以下，得られた解析結果を構文木の形で示す．但し，紙面の都合により，構造の一部を簡略している．

- **例文 (1) の解析結果**

- **例文 (2) の解析結果**


- **例文 (3) の解析結果**



# The Exploration and Analysis of Using Multiple Thesaurus Types for Query Expansion in Information Retrieval

Rila Mandala,† Takenobu Tokunaga† and Hozumi Tanaka†

This paper proposes the use of multiple thesaurus types for query expansion in information retrieval. Hand-crafted thesaurus, corpus-based co-occurrence-based thesaurus and syntactic-relation-based thesaurus are combined and used as a tool for query expansion. A simple word sense disambiguation is performed to avoid misleading expansion terms. Experiments using TREC-7 collection proved that this method could improve the information retrieval performance significantly. Failure analysis was done on the cases in which the proposed method fail to improve the retrieval effectiveness. We found that queries containing negative statements and multiple aspects might cause problems in the proposed method.

**KeyWords:** *multiple thesaurus types, query expansion, information retrieval*

## 1 Introduction

The task of information retrieval system is to extract relevant documents from a large collection of documents in response to user queries (Salton and McGill 1983). Most modern information retrieval systems do not output a set of documents for a query. Instead, they output a list of documents ranked in descending order of relevance to the query (Baeza-Yates and Ribeiro-Neto 1999). In consequence, the task of modern information retrieval system can be re-stated as to push the relevant documents to the top of the retrieved documents rank.

Although information can be presented in diverse form such as tabular numerical data, graphical displays, photographic images, human speech, and so on, the term *information retrieval* as used in this paper shall refer specifically to the retrieval of textual information.

The fundamental problems in information retrieval is that there are many ways to express the same concept in natural language (Blair and Maron 1985; Grossman and Frieder 1998). User in different contexts, or with different information needs or knowledge often describe the same information using different terms. In consequence, relevant document which do not contain the exact terms as the query will be put in low rank.

In this paper, we address the word mismatch problem through automatic query expansion (Ekmekcioglu 1992). The query is expanded by using terms which have related meaning to

† Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

those in the query. The expansion terms can be taken from thesauri (Aitchison and Gilchrist 1987; Paice 1991; Kristensen 1993). Roughly, there are two types of thesauri, i.e., hand-crafted thesauri and corpus-based automatically constructed thesauri. Hand-crafted thesauri describe the synonymous relationship between words, though many thesauri use finer grained relation such as broader terms, narrower terms, and so on. Some of the hand-crafted thesauri are for specific domains while others are for general purpose. The relation in hand-crafted thesauri can be used for query expansion. Query expansion using specific domain thesauri has been reported yielding a very good results (Fox 1980; Chen, Schatz, Yim, and Fye 1995). Currently, the number of existing domain specific thesauri can be counted by finger, while the number of domain in the world is very large. Unfortunately building such thesauri manually requires a lot of human labor from linguists or domain experts and spending very much time. In contrast, the use of general-purpose thesauri for query expansion has been reported fail to improve the information retrieval performance by several researchers (Richardson and Smeaton 1994, 1995; Voorhees 1994, 1988; Smeaton and Berrut 1996; Stairmand 1997).

Automatic thesaurus construction is an extensive studied area in computational linguistics (Charniak 1993; Church and Hanks 1989; Hindle 1990; Lin 1998). The original motivation behind the automatic thesaurus construction is to find an economic alternative to hand-crafted thesaurus. Broadly, there are two methods to construct thesaurus automatically. The first one is based on the similarities between words on the basis of co-occurrence data in each document (Qiu and Frei 1993; Schutze and Pederson 1994, 1997; Crouch 1990; Crouch and Yang 1992), and the other one is based on the co-occurrence of some syntactic relations such as predicate-argument in the whole documents (Jing and Croft 1994; Ruge 1992; Grefenstette 1992; Grafenstette 1994; Hindle 1990).

Many researcher found some slight improvement using the co-occurrence-based thesaurus (Qiu and Frei 1993; Schutze and Pederson 1997), and some mixed results using the syntactic-relation-based thesaurus (Jing and Croft 1994; Grafenstette 1994).

Previously, we conducted an analysis of the different types of thesauri described above, and found that each type of thesaurus has different advantages and disadvantages (Rila Mandala, Tokunaga, Tanaka, Okumura, and Satoh 1999d; Rila Mandala, Tokunaga, and Tanaka 1999c, 1999a, 1999b) which can be summarized as follows :

- Hand-crafted thesaurus
  - can capture general term relation.
  - can not capture domain-specific relation.
- Co-occurrence-based thesaurus

  - can capture domain-specific relation.
  - can not capture the relation between terms which do not co-occur in the same document or window.
- Syntactic-relation-based thesaurus
  - can capture domain-specific relation.
  - can capture the relation between terms even though they do not co-occur in the same document.
  - words with similar heads or modifiers are not always good candidates for expansion

In this paper we explore and analyze a method to combine the three types of thesauri (hand-crafted, co-occurrence-based, and syntactic-relation-based thesaurus) for the purpose of query expansion. In the next section we desribe the detail method of combining thesauri, and in Section 3 we give some experimental results using a large TREC-7 collection and several small information retrieval test collections. We discuss why our method work in Section 4 and also perform failure analysis in Section 5. We tried to combine our method with pseudo-relevance-feedback along with experimental results in Section 6. Finally, in Section 7 we give conclusions and future work.

# 2 Method

In this section, we first describe our method to construct each type of thesaurus utilized in this research, and then describe our attemp to minimize the misleading expansion terms by using term weighting method based on these thesauri.

## 2.1 WordNet

WordNet is a machine-readable hand-crafted thesaurus (Miller 1990). Word forms in WordNet are represented in their familiar orthograpy and word meanings are represented by synonym sets (synset) (Fellbaum 1998). A synonym set represents a concept and comprises all those terms which can be used to express the concept. In other word a synset is a list of synonymous word forms that are interchangeable in some context.

The similarity between words $w_1$ and $w_2$ can be defined as the shortest path from each sense of $w_1$ to each sense of $w_2$, as below (Leacock and Chodorow 1988) :

$$sim_{path}(w_1, w_2) = max[-\log(\frac{N_p}{2D})]$$

where $N_p$ is the number of nodes in path $p$ from $w_1$ to $w_2$ and $D$ is the maximum depth of

the taxonomy.

Similarity also can be measured using the information content of the concepts that subsume words in the taxonomy, as below (Resnik 1995) :

$$sim_{IC}(w_1, w_2) = \max_{c \in S(c_1, c_2)} [-\log p(c)]$$

where $S(c_1, c_2)$ is the set of concepts that subsume both $c_1$ and $c_2$.

Concept probabilities are computed simply as the relative frequency derived from the document collection,

$$p(c) = \frac{freq(c)}{N}$$

where $N$ is the total number of nouns observed, excluding those not subsumed by any WordNet class.

We sum up the path-based similarity and information-content-based similarity to serve as the final similarity.

## 2.2 Co-occurrence-based thesaurus

Co-occurrence-based thesaurus utilize the number of occurrence or co-occurrence of words within a document or within a window as a source of information to build thesaurus. We use textual windows based on TextTiling algorithm (Hearst 1994, 1997) to calculate the mutual information between a pair of words. TextTiling is a paragraph-level model of discourse structure based on the notion of subtopic shift and an algorithm for subdividing expository text into multi-paragraph passages or subtopic segments. This algorithm makes use of patterns of lexical co-occurrence and distribution. The algorithm has three parts: tokenization into terms and sentence-sized units, determination of a score for each sentence-sized unit, and detection of the subtopic boundaries, which are assumed to occur at the largest valleys in the graph that results from plotting sentence-unit against scores. We then employ an information theoretic definition of mutual information which compares the probability of observing two words together to that of observing each word independently in the passages defined by TextTiling. Words having high mutual information over a corpus are assumed semantically related.

## 2.3 Syntactic-relation-based Thesaurus

The basic premise of this method to build thesaurus is that words found in the same grammatical context tend to share semantic similarity. Syntactic analysis allows us to know what words modify other words, and to develop contexts from this information (Grafenstette 1994;

Ruge 1992; Hindle 1990).

To build such thesaurus, firstly, all the documents are parsed using the Apple Pie Parser (Sekine and Grishman 1995). This parser is a bottom-up probabilistic chart parser which finds the parse tree with the best score by way of the best-first search algorithm. Its grammar is a semi-context sensitive grammar with two non-terminals and was automatically extracted from Penn Tree Bank syntactically tagged corpus developed at the University of Pennsylvania. The parser generates a syntactic tree in the manner of a Penn Tree Bank bracketing. The accuracy of this parser is reported as parseval recall 77.45 % and parseval precision 75.58 %.

Using the above parser, we extracted subject-verb, verb-object, adjective-noun, and noun-noun relations, so that each noun has a set of verbs, adjectives, and nouns that it co-occurs with, and for each such relationship, a mutual information value is calculated.

- $I_{sub}(v_i, n_j) = \log \frac{f_{sub}(n_j,v_i)/N_{sub}}{(f_{sub}(n_j)/N_{sub})(f(v_i)/N_{sub})}$
  where $f_{sub}(v_i, n_j)$ is the frequency of noun $n_j$ occurring as the subject of verb $v_i$, $f_{sub}(n_j)$ is the frequency of the noun $n_j$ occurring as subject of any verb, $f(v_i)$ is the frequency of the verb $v_i$, and $N_{sub}$ is the number of subject-verb relations.
- $I_{obj}(v_i, n_j) = \log \frac{f_{obj}(n_j,v_i)/N_{obj}}{(f_{obj}(n_j)/N_{obj})(f(v_i)/N_{obj})}$
  where $f_{obj}(v_i, n_j)$ is the frequency of noun $n_j$ occurring as the object of verb $v_i$, $f_{obj}(n_j)$ is the frequency of the noun $n_j$ occurring as object of any verb, $f(v_i)$ is the frequency of the verb $v_i$, and $N_{obj}$ is the number of verb-object relations.
- $I_{adj}(a_i, n_j) = \log \frac{f_{adj}(n_j,a_i)/N_{adj}}{(f_{adj}(n_j)/N_{adj})(f(a_i)/N_{adj})}$ where $f(a_i, n_j)$ is the frequency of noun $n_j$ occurring as the argument of adjective $a_i$, $f_{adj}(n_j)$ is the frequency of the noun $n_j$ occurring as the argument of any adjective, $f(a_i)$ is the frequency of the adjective $a_i$, and $N_{adj}$ is the number of adjective-noun relations.
- $I_{noun}(n_i, n_j) = \log \frac{f_{noun}(n_j,n_i)/N_{noun}}{(f_{noun}(n_j)/N_{noun})(f(n_i)/N_{noun})}$ where $f(n_i, n_j)$ is the frequency of noun $n_j$ occurring as the argument of noun $n_i$, $f_{noun}(n_j)$ is the frequency of the noun $n_j$ occurring as the argument of any noun, $f(n_i)$ is the frequency of the noun $n_i$, and $N_{noun}$ is the number of noun-noun relations.

The similarity between two words $w_1$ and $w_2$ can be computed as follows :

$$sim(w_1, w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} (I_r(w_1, w) + I_r(w_2, w))}{\sum_{(r,w)\in T(w_1)} I_r(w_1, w) + \sum_{(r,w)\in T(w_2)} I_r(w_2, w)}$$

where $r$ is the syntactic relation type, and $w$ is

- a verb, if $r$ is the subject-verb or object-verb relation.
- an adjective, if $r$ is the adjective-noun relation.

- a noun, if $r$ is the noun-noun relation.

and $T(w)$ is the set of pairs $(r, w')$ such that $I_r(w, w')$ is positive.

## 2.4 Combination and Term Expansion Method

A query $q$ is represented by the vector $\overrightarrow{\mathrm{q}} = (w_1, w_2, ..., w_n)$, where each $w_i$ is the weight of each search term $t_i$ contained in query $q$. We used SMART version 11.0 (Salton 1971) to obtain the initial query weight using the formula $ltc$ as belows :

$$\frac{(\log(tf_{ik}) + 1.0) * \log(N/n_k)}{\sqrt{\sum_{j=1}^{n}[(\log(tf_{ij} + 1.0) * \log(N/n_j)]^2}}$$

where $tf_{ik}$ is the occurrrence frequency of term $t_k$ in query $q_i$, $N$ is the total number of documents in the collection, and $n_k$ is the number of documents to which term $t_k$ is assigned.

Using the above weighting method, the weight of initial query terms lies between 0 and 1. On the other hand, the similarity in each type of thesaurus does not have a fixed range. Hence, we apply the following normalization strategy to each type of thesaurus to bring the similarity value into the range $[0, 1]$.

$$sim_{new} = \frac{sim_{old} - sim_{min}}{sim_{max} - sim_{min}}$$

Although there are many combination methods that can be tried, we just define the similarity value between two terms in the combined thesauri as the average of their similarity value over all types of thesaurus because we do not want to introduce additional parameters here which depend on queries nature.

The similarity between a query $q$ and a term $t_j$ can be defined as belows (Qiu and Frei 1993):

$$simqt(q, t_j) = \sum_{t_i \in q} w_i * sim(t_i, t_j)$$

where the value of $sim(t_i, t_j)$ is taken from the combined thesauri as described above.

With respect to the query $q$, all the terms in the collection can now be ranked according to their $simqt$. Expansion terms are terms $t_j$ with high $simqt(q, t_j)$.

The $weight(q, t_j)$ of an expansion term $t_j$ is defined as a function of $simqt(q, t_j)$:

$$weight(q, t_j) = \frac{simqt(q, t_j)}{\sum_{t_i \in q} w_i}$$

where $0 \leq weight(q, t_j) \leq 1$.

The weight of an expansion term depends both on all terms appearing in a query and on

the similarity between the terms, and ranges from 0 to 1. This weight can be interpreted mathematically as the weighted mean of the similarities between the term $t_j$ and all the query terms. The weight of the original query terms are the weighting factors of those similarities.

Therefore the query $q$ is expanded by adding the following query :

$$\overrightarrow{\mathrm{q_e}} = (a_1, a_2, ..., a_r)$$

where $a_j$ is equal to $weight(q, t_j)$ if $t_j$ belongs to the top $r$ ranked terms. Otherwise $a_j$ is equal to 0.

The resulting expanded query is :

$$\overrightarrow{\mathrm{q}}_{expanded} = \overrightarrow{\mathrm{q}} \circ \overrightarrow{\mathrm{q_e}}$$

where the $\circ$ is defined as the concatenation operator.

The method above can accommodate polysemy, because an expansion term which is taken from a different sense to the original query term is given a very low weight.

# 3 Experimental Results

## 3.1 Test Collection

As a main test collection we use TREC-7 collection (Voorhees and Harman 1999). TREC (Text REtrieval Conference) is an DARPA (Defense Advanced Research Project Agency) and NIST (National Institute of Standards and Technology) co-sponsored effort that brings together information retrieval researchers from around the world to discuss and compare the performance of their systems, and to develop a large test collection for information retrieval system. The seventh in this series of annual conferences, TREC-7, attracted 56 different participants from academic institutions, government organizations, and commercial organizations (Voorhees and Harman 1999). With such a large participation of various information retrieval researchers, a large and varied collections of full-text documents, a large number of user queries, and a superior set of independent relevance judgements, TREC collections have rightfully become the standard test collections for current information retrieval research.

The common information retrieval task of ranking documents for a new query is called the *adhoc* task in the TREC framework. The TREC data comes on CD-ROMs, called the TREC disks. The disks are numbered, and a combination of several disk can be used to form a text collection for experimentation.

The TREC-7 test collection consists of 50 topics (queries) and 528,155 documents from

**Table 1** TREC-7 Document statistics

| Source | Size (Mb) | Number of documents | Average number of terms/article |
|---|---|---|---|
| **Disk 4** | | | |
| The Financial Times, 1991-1994 (FT) | 564 | 210,158 | 412.7 |
| Federal Register, 1994 (FR94) | 395 | 55,630 | 644.7 |
| **Disk 5** | | | |
| Foreign Broadcast Information Services (FBIS) | 470 | 130,471 | 543.6 |
| the LA Times | 475 | 131,896 | 526.5 |

several sources: the Financial Times (FT), Federal Register (FR94), Foreign Broadcast Information Service (FBIS) and the LA Times. Each topic consists of three sections, the *Title*, *Description* and *Narrative*. Table 1 shows statistics of the TREC-7 document collection, Table 2 shows statistics of the topics, and Figure 1 shows an example of a topic, and Figure 2 shows its expansion terms produced by our method.

**Table 2** TREC-7 topic length statistics (words)

| Topic section | Min | Max | Mean |
|---|---|---|---|
| Title | 1 | 3 | 2.5 |
| Description | 5 | 34 | 14.3 |
| Narrative | 14 | 92 | 40.8 |
| All | 31 | 114 | 57.6 |

**Title:**
clothing sweatshops

**Description:**
Identify documents that discuss clothing sweatshops.

**Narrative:**
A relevant document must identify the country, the working conditions, salary, and type of clothing or shoes being produced. Relevant documents may also include the name of the business or company or the type of manufacturing, such as: "designer label".

**Fig. 1** Topics Example

| | | | | | |
|---|---|---|---|---|---|
| wage | labor | sewing | low | minimum | payment |
| earning | workshop | workplace | shop | welfare | county |
| circumstance | overtime | child | entrepreneur | employment | manufacture |
| immigrant | industry | bussiness | company | violation | remuneration |
| apparel | vesture | wear | footwear | footgear | enterprise |
| commercialism | machine | status | plant | raise | production |
| calcitonin | | | | | |

**Fig. 2** Expansion terms example

It is well known that many information retrieval techniques are sensitive to factors such as query length, document length, and so forth. For example, one technique which works very well for long queries may not work well for short queries. To ensure that our techniques and conclusions are general, we use different-length query in TREC-7 collection.

Beside the large and the newer TREC-7 test collection described before, we also use some previous small test collections (Fox 1990), because although most real world collections are large, some can be quite small. These small collections have been widely used in the experiments by many information retrieval researchers before TREC. These old test collections have always been built to serve some purpose. For example, the Cranfield collection was originally built to test different types of manual indexing, the MEDLINE collection was built in an early attempt to compare the operational Boolean MEDLARS system with the experimental ranking used in SMART, and the CACM and CISI collections were built to investigate the use of an extended vector space model that included bibliographic data. Most of the old test collections are very domain specific and contain only the abstract.

In Table 3 and 4 we describe the statistics and the domain of the old collection, respectively.

## 3.2 Evaluation method

Recall and precision are two widely used metrics to measure the retrieval effectiveness of an information retrieval system. Recall is the fraction of the relevant documents which has been retrieved, i.e.

$$recall = \frac{number\ of\ relevant\ documents\ retrieved}{number\ of\ relevant\ documents\ in\ collection}.$$

**Table 3** Small collection statistics

| Collection | Number of Documents | Average Terms/Docs | Number of Query | Average Terms/query | Average Relevant/query |
|---|---|---|---|---|---|
| Cranfield | 1398 | 53.1 | 225 | 9.2 | 7.2 |
| ADI | 82 | 27.1 | 35 | 14.6 | 9.5 |
| MEDLARS | 1033 | 51.6 | 30 | 10.1 | 23.2 |
| CACM | 3204 | 24.5 | 64 | 10.8 | 15.3 |
| CISI | 1460 | 46.5 | 112 | 28.3 | 49.8 |
| NPL | 11429 | 20.0 | 100 | 7.2 | 22.4 |
| INSPEC | 12684 | 32.5 | 84 | 15.6 | 33.0 |

**Table 4** The domain of the small collections

| Collection | Domain |
|---|---|
| Cranfield | Aeronautics |
| ADI | Information Science |
| MEDLINE | Medical Science |
| CACM | Computer Science |
| CISI | Computer and Information Science |
| NPL | Electrical Engineering |
| INSPEC | Electrical Engineering |

Precision is the fraction of the retrieved document, i.e.

$$precision = \frac{number\ of\ relevant\ documents\ retrieved}{total\ number\ of\ documents\ retrieved}.$$

However, precision and recall are set-based measures. That is, they evaluate the quality of an unordered set of retrieved documents. To evaluate ranked lists, precision can be plotted against recall after each retrieved document. To facilitate comparing performance over a set of topics, each with a different number of relevant documents, individual topic precision values are interpolated to a set of standard recall levels (0 to 1 in increments of 0.1). The particular rule used to interpolate precision at standard recall level $i$ is to use the maximum precision obtained for the topic for any actual recall level greater than or equal to $i$. Note that while precision is not defined at a recall 0.0, this interpolation rule does define an interpolated value for recall level 0.0. For example assume a document collection has 20 documents, four of which are relevant to topic $t$ in which they are retrieved at ranks 1, 2, 4, 15. The exact recall points are 0.25, 0.5, 0.75, and 1.0. Using the interpolation rule, the interpolated precision for all standard recall levels 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5 is 1, the interpolated precision for recall levels 0.6 and 0.7 is 0.75, and the interpolated precision for recall levels 0.8, 0.9, and 1.0 is

0.27.

### 3.3 Results

Table 5 shows the average of 11-point interpolated precision using various section of topics in TREC-7 collection, and Table 6 shows the average of 11-point interpolated precision in several small collections. We can see that our method give a consistent and significant improvement compared with the baseline and using only one type of thesaurus.

**Table 5** Experiment results using TREC-7 Collection

| Topic Type | Base | Expanded with | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | WordNet only | Syntactic only | Cooccur only | WordNet+ Syntactic | WordNet+ Cooccur | Syntactic+ Cooccur | Combined method |
| Title | 0.1452 | 0.1541 (+6.1%) | 0.1802 (+24.1%) | 0.1905 (+31.2%) | 0.1877 (+29.3%) | 0.2063 (+42.1%) | 0.2197 (+51.3%) | 0.2659 (+83.1 %) |
| Description | 0.1696 | 0.1777 (+4.8%) | 0.1974 (+16.4%) | 0.2144 (+26.4%) | 0.2057 (+21.3%) | 0.2173 (+28.1%) | 0.2337 (+37.8%) | 0.2722 (+60.5 %) |
| All | 0.2189 | 0.2235 (+2.1%) | 0.2447 (+11.8%) | 0.2566 (+17.2%) | 0.2563 (+17.1%) | 0.2611 (+19.3%) | 0.2679 (+22.4%) | 0.2872 (+31.2 %) |

**Table 6** Experiment results using small collection

| Coll | Base | Expanded with | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | WordNet only | Syntactic only | Cooccur only | WordNet+ Syntactic | WordNet+ Cooccur | Syntactic+ Cooccur | Combined method |
| ADI | 0.4653 | 0.4751 (+2.1%) | 0.5039 (+8.3%) | 0.5146 (+10.6%) | 0.5263 (+13.1%) | 0.5486 (+17.9%) | 0.5895 (+26.7%) | 0.6570 (+41.2%) |
| CACM | 0.3558 | 0.3718 (+4.5%) | 0.3853 (+8.3%) | 0.4433 (+24.6%) | 0.4109 (+15.5%) | 0.4490 (+26.2%) | 0.4796 (+34.8%) | 0.5497 (+54.5%) |
| INSPEC | 0.3119 | 0.3234 (+3.7%) | 0.3378 (+8.3%) | 0.3755 (+20.4%) | 0.3465 (+11.1%) | 0.4002 (+28.3%) | 0.4420 (+41.7%) | 0.5056 (+62.1 %) |
| CISI | 0.2536 | 0.2719 (+7.2%) | 0.2800 (+10.4%) | 0.3261 (+28.6%) | 0.3076 (+21.3%) | 0.3606 (+42.2%) | 0.4009 (+58.1%) | 0.4395 (+73.3 %) |
| CRAN | 0.4594 | 0.4700 (+2.3%) | 0.4916 (+7.0%) | 0.5435 (+18.3%) | 0.5012 (+9.1%) | 0.5706 (+24.2%) | 0.5931 (+29.1%) | 0.6528 (+42.1 %) |
| MEDLINE | 0.5614 | 0.5681 (+1.2%) | 0.6013 (+7.1%) | 0.6372 (+13.5%) | 0.6114 (+8.9%) | 0.6580 (+17.2%) | 0.6860 (+22.2%) | 0.7551 (+34.5%) |
| NPL | 0.2700 | 0.2840 (+5.2%) | 0.2946 (+9.1%) | 0.3307 (+22.5%) | 0.3038 (+12.5%) | 0.3502 (+29.7%) | 0.3796 (+40.6%) | 0.4469 (+65.5%) |

## 4 Discussion

The important points of our method are :

- the coverage of WordNet is broadened
- weighting method.

The three types of thesauri we used have different characteristics. Automatically constructed thesauri add not only new terms but also new relationships not found in WordNet. If two terms often co-occur together in a document then those two terms are likely bear some

relationship. Why not only use the automatically constructed thesauri ? The answer to this is that some relationships may be missing in the automatically constructed thesauri (Grafenstette 1994). For example, consider the words *tumor* and *tumour*. These words certainly share the same context, but would never appear in the same document, at least not with a frequency recognized by a co-occurrence-based method. In general, different words used to describe similar concepts may never be used in the same document, and are thus missed by the co-occurrence methods. However their relationship may be found in the WordNet thesaurus.

The second point is our weighting method. As already mentioned before, most attempts at automatically expanding queries by means of WordNet have failed to improve retrieval effectiveness. The opposite has often been true: expanded queries were less effective than the original queries. Beside the "incomplete" nature of WordNet, we believe that a further problem, the weighting of expansion terms, has not been solved. All weighting methods described in the past researches of query expansion using WordNet have been based on "trial and error" or ad-hoc methods. That is, they have no underlying justification.

The advantages of our weighting method are:

- the weight of each expansion term considers the similarity of that term with all terms in the original query, rather than to just one or some query terms.
- the weight of the expansion term accommodates the polysemous word problem.

This method can accommodate the polysemous word problem, because an expansion term taken from a different sense to the original query term sense is given very low weight. The reason for this is that, the weighting method depends on all query terms and all of the thesauri. For example, the word *bank* has many senses in WordNet. Two such senses are the financial institution and the river edge senses. In a document collection relating to financial banks, the river sense of *bank* will generally not be found in the co-occurrence-based thesaurus because of a lack of articles talking about rivers. Even though (with small possibility) there may be some documents in the collection talking about rivers, if the query contained the finance sense of *bank* then the other terms in the query would also concerned with finance and not rivers. Thus rivers would only have a relationship with the *bank* term and there would be no relationships with other terms in the original query, resulting in a low weight. Since our weighting method depends on both query in its entirety and similarity in the three thesauri, the wrong sense expansion terms are given very low weight.

# 5 Failure Analysis

Although our method as a whole gives a very significant improvement, it still further can be improved. Of the 50 queries of TREC-7 collection, our method improves the performance of 43 queries and degrade the performance of 7 queries compared with the baseline. We investigated manually why our method degrade the performance of several queries.

## 5.1 Negation statements in the query

We found that most of the queries hurted by our method contains the negation statements. Through our method, all the terms in the negation statements are also considered for query expansion which is degrading the retrieval performance for that query. Figure 3 shows two examples of query which contain negation statements.

Table 7 shows the results of eliminating the negation statements from the queries manually for each query containing negation statements. As that table shown, eliminating the negation statements improves the retrieval effectiveness. It is to be investigated further how we could identify the negation statements automatically.

**Table 7** The results of negation statements elimination

| Query Number | SMART | Expansion without Negation Elimination | Expansion with Negation Elimination |
|---|---|---|---|
| 2 | 0.3643 | 0.3381 (- 7.19%) | 0.3811 (+ 4.61%) |
| 5 | 0.3112 | 0.2804 (- 9.90%) | 0.3314 (+ 6.49%) |
| 13 | 0.1621 | 0.1567 (- 3.33%) | 0.1823 (+12.46%) |
| 17 | 0.2310 | 0.2235 (- 3.25%) | 0.2441 (+ 5.67%) |
| 42 | 0.2732 | 0.2569 (- 5.97%) | 0.2942 (+ 7.69%) |
| 43 | 0.3031 | 0.2834 (- 6.50%) | 0.3321 (+ 9.57%) |

## 5.2 Multiple aspects of query

An examination of the top-ranked non-relevant documents for various queries shows that a commonly occurring cause of non-relevance among such documents is inadequate query

**Title:**
British Chunnel impact

**Description:**
What impact has the Chunnel had on the British economy and/or the life style of the British?

**Narrative:**
Documents discussing the following issues are relevant:
- projected and actual impact on the life styles of the British
- Long term changes to economic policy and relations
- major changes to other transportation systems linked with the Continent

Documents discussing the following issues are not relevant:
- expense and construction schedule
- routine marketing ploys by other channel crossers (i.e., schedule changes, price drops, etc.)

**Title:**
Ocean remote sensing

**Description:**
Identify documents discussing the development and application of spaceborne ocean remote sensing.

**Narrative:**
Documents discussing the development and application of spaceborne ocean remote sensing in oceanography, seabed prospecting and mining, or any marine-science activity are relevant. Documents that discuss the application of satellite remote sensing in geography, agriculture, forestry, mining and mineral prospecting or any land-bound science are not relevant, nor are references to international marketing or promotional advertizing of any remote-sensing technology. Synthetic aperture radar (SAR) employed in ocean remote sensing is relevant.

**Fig. 3** Two examples of query containing negation statements

coverage, i.e., the query consists of multiple aspects, only some of which are covered in these documents. For example, a query of the TREC collection asks : *Identify documents discussing the use of estrogen by postmenopausal women in Britain.* Several top-ranked non-relevant documents contain information about the use of hormone by postmenopausal women but not in Britain. If we look at the expansion terms produced by our method as shown in Figure 4 we could see that many expansion terms have relationship with all query terms except Britain. This is because all query terms but Britain have relationship between each other and these terms have a high original term weight. On the contrary, Britain does not have relationship with other query terms and Britain have a low original term weight in almost all documents in collection. Consequently, the term related to Britain are given a low weight by our method.

| | | | | | |
|---|---|---|---|---|---|
| estradiol | female | hormone | disease | therapy | menopausal |
| chemical | progesterone | menstruation | vaginal | progestin | obstetrics |
| gynecology | replacement | endometrial | cancer | breast | ovary |
| treatment | old | tamoxifen | symptom | synthetic | drug |
| hot | flash | osteoporosis | cholesterol | recepter | risk |
| calcium | bones | mineralization | medical | physiologist | diagnostic |
| calcitonin | | | | | |

**Fig. 4** Expansion terms

To investigate the relatedness or independence of query words, we examine their co-occurrence patterns in 1000 documents initially retrieved for a query. If two words have the same aspect, then they often occur together in many of these documents. If one of the words appears in a document, the chance of the other occurring within the same document is likely to be relatively high. On the other hand, if two words bear independent concepts, the occurrences of the words are not strongly related.

Based on this observation, we re-rank the top-1000 retrieved documents, by re-computing the similarity between a query $\overrightarrow{q} = \{t_1, t_2, .., t_m\}$ (terms are ordered by decreasing of their inverse document frequency) and document $D$ as belows (Mitra, Singhal, and Buckley 1998) :

$$Sim_{new}(D) = idf(t_1) + \sum_{i=2}^{m} idf(t_i) \ \times \ min_{j=1}^{i-1}(1 - P(t_i|t_j)),$$

where $idf$ is the inverse of document frequency in the top-1000 initially retrieved documents, $m$ is the number of terms in query that appear in document $D$, and $P(t_i|t_j)$ is estimated based

on word occurrences in document collection and is given by :

$$\frac{\#\ documents\ containing\ words\ t_i\ and\ t_j}{\#\ documents\ containing\ word\ t_j}.$$

For example, in the query stated above, the terms *estrogen, postmenopausal,* and *women* are strongly related to each other. If the term *postmenopausal* occurs in a document, the probability of word *women* occurring in the same document is high. Accordingly, the contribution of word *women* to $Sim_{new}$ is reduced in this case. On the other hand, terms *postmenopausal* and *Britain* correspond to two independent aspects of the query and the occurrences of these two terms are relatively uncorrelated. Therefore, if a document contains these two terms, the contribution of *Britain* is higher and it counts as an important new matching term since its occurrence is not well predicted by other matching term (*postmenopausal*). This technique can improve the average of 11-point interpolated precision of TREC-7 collection for about 3.3% as shown in Table 8.

We also investigated another method to overcome this problem in which we built a Boolean expression for all query manually. Terms in the same aspect of query are placed in *or* relation, and terms in different aspect are placed in *and* relation (Hearst 1996). Documents that satisfy the constraint contain at least one word from each aspect of the query. For example, for the query stated before (*Identify documents discussing the use of estrogen by postmenopausal women in Britain*), we construct boolean expression as follows :

```
estrogen and (postmenopausal or woman) and britain.
```

Using this method, we again re-rank the top 1000 documents initially retrieved. Documents that match more words in different aspect of query are ranked ahead of documents that match less words. Ties are resolved by referring to the original document weight. Using this method we can improve the average of 11-point interpolated precision of TREC-7 collection for about 11.3%, as shown in Table 8.

This correlation and boolean reranking methods degrade some queries performance, because in those queries these methods overweight several query terms.

It is to be further investigated how we could design the appropriate method to overcome this problem.

# 6 Combining with relevance feedback

In this section, we describe the combination of our method with pseudo-relevance feedback (Buckley and Salton 1994, 1995; Salton and Buckley 1990). Pseudo-relevance feedback

**Table 8** The effect of re-ranking the top-1000 ranked initially retrieved using co-occurrence method and boolean filter method

| Query Number | Without Re-ranking | Re-ranking correlation | %improvement | Reranking Boolean | %improvement |
|---|---|---|---|---|---|
| 1 | 0.5153 | 0.5666 | +9.96 | 0.7724 | +49.89 |
| 2 | 0.3794 | 0.1952 | -48.55 | 0.4740 | +24.93 |
| 3 | 0.3230 | 0.2719 | -15.82 | 0.3237 | +0.22 |
| 4 | 0.2280 | 0.2731 | +19.78 | 0.2355 | +3.29 |
| 5 | 0.3213 | 0.2457 | -23.53 | 0.2931 | -8.78 |
| 6 | 0.0646 | 0.0495 | -23.37 | 0.0655 | +1.39 |
| 7 | 0.3878 | 0.5632 | +45.23 | 0.3607 | -6.99 |
| 8 | 0.2983 | 0.4270 | +43.14 | 0.3049 | +2.21 |
| 9 | 0.0422 | 0.0612 | +45.02 | 0.0254 | -39.81 |
| 10 | 0.2196 | 0.3223 | +46.77 | 0.3619 | +64.80 |
| 11 | 0.5802 | 0.3524 | -39.26 | 0.4950 | -14.68 |
| 12 | 0.3588 | 0.1466 | -59.14 | 0.2319 | -35.37 |
| 13 | 0.1745 | 0.0908 | -47.97 | 0.0868 | -50.26 |
| 14 | 0.6055 | 0.5604 | -7.45 | 0.4963 | -18.03 |
| 15 | 0.8877 | 0.9451 | +6.47 | 0.8554 | -3.64 |
| 16 | 0.3856 | 0.3094 | -19.76 | 0.4823 | +25.08 |
| 17 | 0.2360 | 0.1363 | -42.25 | 0.1479 | -37.33 |
| 18 | 0.7882 | 0.6419 | -18.56 | 0.6662 | -15.48 |
| 19 | 0.5141 | 0.4027 | -21.67 | 0.4177 | -18.75 |
| 20 | 0.1871 | 0.3997 | +113.63 | 0.3016 | +61.20 |
| 21 | 0.0152 | 0.0346 | +127.63 | 0.0837 | +450.66 |
| 22 | 0.0920 | 0.3644 | +296.09 | 0.1399 | +52.07 |
| 23 | 0.2328 | 0.4043 | +73.67 | 0.4277 | +83.72 |
| 24 | 0.3250 | 0.3177 | -2.25 | 0.3951 | +21.57 |
| 25 | 0.5943 | 0.2812 | -52.68 | 0.3239 | -45.50 |
| 26 | 0.2360 | 0.2312 | -2.03 | 0.1034 | -56.19 |
| 27 | 0.4634 | 0.3062 | -33.92 | 0.3322 | -28.31 |
| 28 | 0.0307 | 0.0306 | -0.33 | 0.0142 | -53.75 |
| 29 | 0.0314 | 0.2575 | +720.06 | 0.3349 | +966.56 |
| 30 | 0.2162 | 0.2164 | +0.09 | 0.3832 | +77.24 |
| 31 | 0.0500 | 0.0560 | +12.00 | 0.0635 | +27.00 |
| 32 | 0.4544 | 0.5968 | +31.34 | 0.5803 | +27.71 |
| 33 | 0.0220 | 0.0232 | +5.45 | 0.0290 | +31.82 |
| 34 | 0.2169 | 0.1989 | -8.30 | 0.2299 | + 5.99 |
| 35 | 0.2267 | 0.3421 | +50.90 | 0.4012 | +76.97 |
| 36 | 0.0129 | 0.0286 | +121.71 | 0.0406 | +214.73 |
| 37 | 0.2563 | 0.2605 | +1.64 | 0.2289 | -10.69 |
| 38 | 0.2534 | 0.2300 | -9.23 | 0.2079 | -17.96 |
| 39 | 0.0006 | 0.0200 | +3233.33 | 0.0085 | +1316.67 |
| 40 | 0.2004 | 0.3230 | +61.18 | 0.2708 | +35.13 |
| 41 | 0.0015 | 0.4938 | +32820.00 | 0.5261 | +34973.33 |
| 42 | 0.2883 | 0.1346 | -53.31 | 0.4216 | +46.24 |
| 43 | 0.2996 | 0.1280 | -57.28 | 0.1684 | -43.79 |
| 44 | 0.0218 | 0.1019 | +367.43 | 0.0952 | +336.70 |
| 45 | 0.1506 | 0.1879 | +24.77 | 0.2783 | +84.79 |
| 46 | 0.3485 | 0.6087 | +74.66 | 0.4719 | +35.41 |
| 47 | 0.0967 | 0.0303 | -68.67 | 0.3293 | +240.54 |
| 48 | 0.3886 | 0.3418 | -12.04 | 0.2954 | -23.98 |
| 49 | 0.2066 | 0.1351 | -34.61 | 0.1826 | -11.62 |
| 50 | 0.3861 | 0.4312 | +11.68 | 0.3978 | +3.03 |
| Average | 0.2723 | 0.2815 | +3.3 | 0.3033 | +11.3 |

is a feedback aproach without requiring relevance information. Instead, an initial retrieval is performed, and the top-$n$ ranked documents are all assumed to be relevant for obtaining expansion terms ($\overrightarrow{\mathrm{q}}_{feedback}$) as belows :

$$\overrightarrow{\mathrm{q}}_{feedback} = \frac{1}{|D_r|} \sum_{d_i \in D_r} \overrightarrow{d_i}$$

In this case, $D_r$ is a set of documents ranked on the top in the initial retrieval and $\overrightarrow{d_i}$ is the vector representation of document $d_i$.

In the framework of the inference network (Xu and Croft 1996), the information need of the user is represented by multiple queries. Multiple queries means that an information need is represented by some different query representation. Experiments show that multiple query representations can produce better results than using one representation alone. However, how

to obtain these queries is not discussed in this model. Hence we try to find multiple query representations for the information structure derived from feedback information. In this way, the following three representations can be obtained :

- representation derived directly from the original query : $\overrightarrow{\mathrm{q}}_{original}$,
- representation obtained by our method : $\overrightarrow{\mathrm{q}}_{thesauri}$,
- representation derived from the retrieved documents of the previous run : $\overrightarrow{\mathrm{q}}_{feedback}$.

A linear combination of the three query representations is used to retrieve documents. However, we do not introduce additional parameters which are quite difficult to determine. Also we believe that the parameter values determined for some queries may not be suitable for some other queries because they are query dependent. Hence the simple combination we use is :

$$\overrightarrow{\mathrm{q}}_{original} + \overrightarrow{\mathrm{q}}_{thesauri} + \overrightarrow{\mathrm{q}}_{feedback}.$$

When using the relevance-feedback method, we used the top 30 ranked documents of the previous run of the original query to obtain $\overrightarrow{\mathrm{q}}_{feedback}$.

In order to evaluate the retrieval effectiveness of the new method, we carried out some experiments using TREC-7 collection to compare the retrieval effectiveness of the following methods using different combination of the query representations. Figure 5 shows 11-point interpolated precision using our method alone, pseudo-feedback alone, and the combination of our method and pseudo-feedback. Our method alone has better performance than the pseudo-feedback method, and the combination of our method and pseudo-feedback slightly better than our method alone.

Recently, Xu and Croft (1996) suggested a method called local context analysis, which also utilize the co-occurrence-based thesaurus and relevance feedback method. Instead of gathering co-occurrence data from the whole corpus, he gather it from the top-$n$ ranked document. We carry out experiments in that we build the combined-thesauri based on the top-$n$ ranked document, rather than the whole corpus. As can be seen in Figure 6, query expansion using the combined thesauri built from the top-$n$ ranked document have a lower performance than query expansion using the combined thesauri built from the whole corpus.

# 7 Conclusions and Future Work

We have proposed the use of multiple types of thesauri for query expansion in information retrieval, give some failure analysis, and combining our method with pseudo-relevance feedback method. The basic idea underlying our method is that each type of thesaurus has

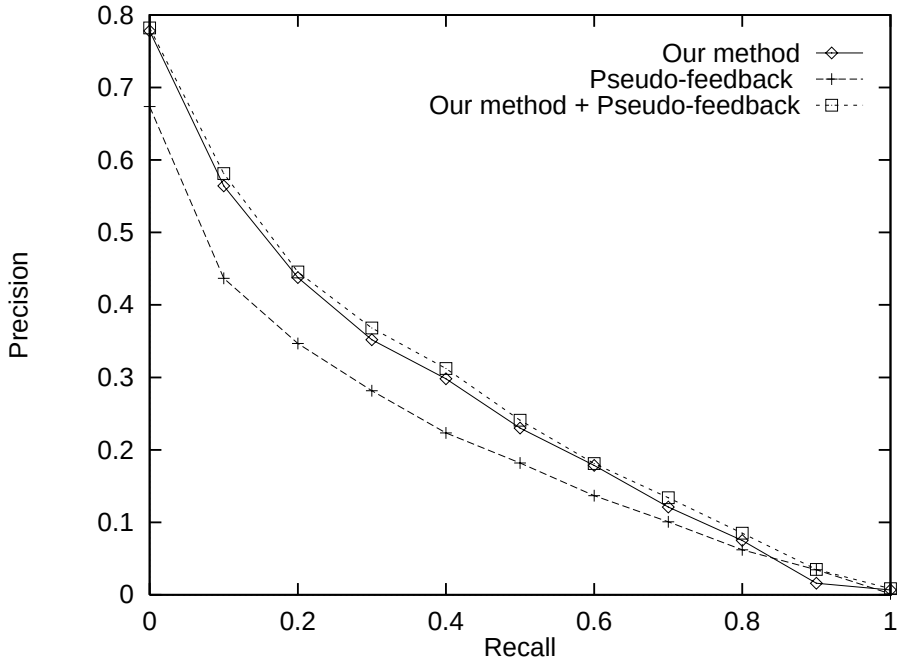



**Fig. 5** The results of combining our method and pseudo-feedback

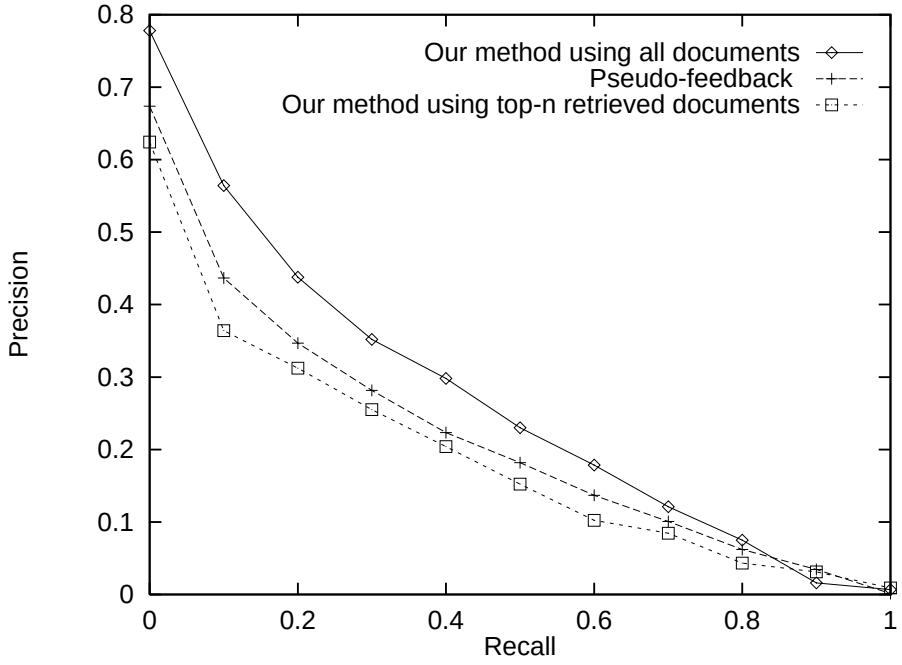



**Fig. 6** The results of combined thesauri built from the top-$n$ ranked document

different characteristics and combining them provides a valuable resource to expand the query. Misleading expansion terms can be avoided by designing a weighting term method in which the weight of expansion terms not only depends on all query terms, but also depends on their similarity values in all type of thesaurus.

Future research will include the use of parser with better performance, designing a general algorithm for automatically handling the negation statements, and also designing an effective algorithm for handling the multiple aspect contain in the query.

# 8 Acknowledgments

The authors would like to thank the anonymous referees for useful comments on the earlier version of this paper. We also thank Chris Buckley (SabIR Research) for support with SMART, Satoshi Sekine (New York University) for the Apple Pie Parser, Akitoshi Okumura (NEC C & C Media Lab.) for providing the computer environments in very preliminary experiments. This research is partially supported by JSPS project number JSPS-RFTF96P00502.

# Reference

Aitchison, J. and Gilchrist, A. (1987). *Thesaurus Construction A Practical Manual*. Aslib.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addisson Wesley.

Blair, D. and Maron, M. (1985). "An evaluation of retrieval effectiveness." *Communications of the ACM*, *28*, 289–299.

Buckley, C. and Salton, G. (1994). "The Effect of Adding Relevance Information in a Relevance Feedback Environment." In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval Conference*, pp. 292–300.

Buckley, C. and Salton, G. (1995). "Automatic Query Expansion using SMART: TREC-3." In *Proceedings of The Third Text Retrieval Conference*, pp. 69–80.

Charniak, E. (1993). *Statistical Language Learning*. MIT Press.

Chen, H., Schatz, B., Yim, T., and Fye, D. (1995). "Automatic Thesaurus Generation for an Electronic Community System." *Journal of American Society for Information Science*, *46*(3), 175–193.

Church, K. and Hanks, P. (1989). "Word Association Norms,Mutual Information and Lexicography." In *Proceedings of the 27nd Annual Meeting of the Association for Computational Linguistics*, pp. 76–83.

Crouch, C. J. (1990). "An Aproach to The Automatic Construction of Global Thesauri." *Information Processing and Management*, *26*(5), 629–640.

Crouch, C. and Yang, B. (1992). "Experiments in Automatic Statistical Thesaurus Construction." In *Proceedings of the Fifteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval Conference*, pp. 77–82.

Ekmekcioglu, F. (1992). "Effectiveness of Query Expansion in Ranked-Output Document Retrieval Systems." *Journal of Information Science*, *18*, 139–147.

Fellbaum, C. (1998). *WordNet,An Electronic Lexical Database*. MIT Press.

Fox, E. A. (1980). "Lexical Relations Enhancing Effectiveness of Information Retrieval Systems." *SIGIR Forum*, *15*(3), 6–36.

Fox, E. A. (1990). *Virginia Disk One*. Blacksburg: Virginia Polytechnic Institute and State University.

Grafenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publisher.

Grefenstette, G. (1992). "Use of Syntactic Context to Produce Term Association Lists for Text Retrieval." In *Proceedings of the 15th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval Conference*, pp. 89–97.

Grossman, D. and Frieder, O. (1998). *Information Retrieval: Algorithms and Heuristics*. Kluwer Academic Publishers.

Hearst, M. A. (1994). "Multi-Paragraph Segmentation of Expository Text." In *Proceedings of 32th Annual Meeting of the Association for Computational Linguistics*, pp. 9–16.

Hearst, M. A. (1996). "Improving Full-Text Precision on Short Queries using Simple Constraints." In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*.

Hearst, M. A. (1997). "TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages." *Computational Linguistics*, *23*(1), 33–64.

Hindle, D. (1990). "Noun Classification from Predicate-Argument Structures." In *Proceedings of 28th Annual Meeting of the Association for Computational Linguistics*, pp. 268–275.

Jing, Y. and Croft, B. (1994). "An Association Thesaurus for Information Retrieval." In *Proceedings of RIAO*, pp. 146–160.

Kristensen, J. (1993). "Expanding End-Users Query Statements for Free Text Searching with a Search-Aid Thesaurus." *Information Processing and Management*, *29*(6), 733–744.

Leacock, C. and Chodorow, M. (1988). "Combining Local Context and WordNet Similarity for Word Sense Identification." In Fellbaum, C. (Ed.), *WordNet,An Electronic Lexical*

*Database*, pp. 265–283. MIT Press.

Lin, D. (1998). "Automatic Retrieval and Clustering of Similar Words." In *Proceedings of the COLING-ACL'98*, pp. 768–773.

Miller, G. (1990). "WordNet: An On-line Lexical Database." *Special Issue of the International Journal of Lexicography*, *3*(4).

Mitra, M., Singhal, A., and Buckley, C. (1998). "Improving Automatic Query Expansion." In *Proceedings of the 21th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 206–214.

Paice, C. D. (1991). "A Thesaural Model of Information Retrieval." *Information Processing and Management*, *27*(5), 433–447.

Qiu and Frei, H. (1993). "Concept Based Query Expansion." In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval Conference*, pp. 160–169.

Resnik, P. (1995). "Using Information Content to Evaluate Semantic Similarity in a Taxonomy." In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 448–453.

Richardson, R. and Smeaton, A. (1994). "Using WordNet for Conceptual Distance Measurement." In *Proceedings of the BCS-IRSG Colloqium.*

Richardson, R. and Smeaton, A. F. (1995). "Using WordNet in a Knowledge-Based Approach to Information Retrieval." Tech. rep. CA-0395, School of Computer Applications,Dublin City University.

Rila Mandala, Tokunaga, T., and Tanaka, H. (1999a). "Combining General Hand-Made and Automatically Constructed Thesauri for Information Retrieval." In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pp. 920–925.

Rila Mandala, Tokunaga, T., and Tanaka, H. (1999b). "Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion." In *Proceedings of the 22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 191–197.

Rila Mandala, Tokunaga, T., and Tanaka, H. (1999c). "Complementing WordNet with Roget and Corpus-based Thesauri for Information Retrieval." In *Proceedings of the 9th European Chapter of the Association for Computational Linguistics (EACL'99)*, pp. 94–101.

Rila Mandala, Tokunaga, T., Tanaka, H., Okumura, A., and Satoh, K. (1999d). "Adhoc Retrieval Experiments Using WordNet and Automatically Constructed Thesauri." In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pp. 475–480. NIST

Special Publication.

Ruge, G. (1992). "Experiments on Linguistically-Based Term Associations." *Information Processing and Management*, *28*(3), 317–332.

Salton, G. (1971). *The SMART Retrieval System Experiments in Automatic Document Processing*. Prentice-Hall.

Salton, G. and Buckley, C. (1990). "Improving Retrieval Performance by Relevance Feedback." *Journal of American Society for Information Science*, *41*(4), 288–297.

Salton, G. and McGill, M. (1983). *An Introduction to Modern Information Retrieval*. McGraw-Hill.

Schutze, H. and Pederson, J. (1994). "A Cooccurrence-Based Thesaurus and Two Applications to Information Retrieval." In *Proceedings of RIAO Conference*, pp. 266–274.

Schutze, H. and Pederson, J. (1997). "A Cooccurrence-Based Thesaurus and Two Applications to Information Retrieval." *Information Processing and Management*, *33*(3), 307–318.

Sekine, S. and Grishman, R. (1995). "A Corpus-based Probabilistic Grammar with Only Two Non-terminals." In *Proceedings of the International Workshop on Parsing Technologies*.

Smeaton, A. and Berrut, C. (1996). "Thresholding Postings Lists,Query Expansion by Word-Word Distances and POS Tagging of Spanish Text." In *Proceedings of The Fourth Text Retrieval Conference*.

Stairmand, M. (1997). "Textual Context Analysis for Information Retrieval." In *Proceedings of the 20th ACM-SIGIR Conference*, pp. 140–147.

Voorhees, E. M. (1988). "Using WordNet for Text Retrieval." In Fellbaum, C. (Ed.), *WordNet,An Electronic Lexical Database*, pp. 285–303. MIT Press.

Voorhees, E. (1994). "Query Expansion using Lexical-Semantic Relations." In *Proceedings of the 17th ACM-SIGIR Conference*, pp. 61–69.

Voorhees, E. and Harman, D. (1999). "Overview of the Seventh Text retrieval Conference (TREC-7)." In *Proceedings of the Seventh Text REtrieval Conference*. NIST Special Publication.

Xu, J. and Croft, B. (1996). "Query Expansion Using Local and Global Document Analysis." In *Proceedings of the 19th ACM-SIGIR Conference*, pp. 4–11.

**Rila Mandala:** He is a lecturer in Department of Informatics, Bandung Institute of Technology, Indonesia since 1992. He received the B.S. degree in informatics from Bandung Institute of Technology, Indonesia and M.Eng. degree in computer science from Tokyo Institute of Technology, Japan, in

1992 and 1996, respectively. Currently, he is a doctoral student of Department of Computer Science, Tokyo Institute of Technology. His current research interests are information retrieval, computational linguistics, and natural language processing.

**Takenobu Tokunaga:** He is an associate professor of Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He received the B.S. degree in 1983 from Tokyo Institute of Technology, the M.S and the Dr.Eng. degrees from Tokyo Institute of Technology in 1985 and 1991, respectively. His current interests are computational linguistics and information retrieval.

**Hozumi Tanaka:** He is a professor of Department of Computer Science, Tokyo Institute of Technology. He received the B.S. degree in 1964 and the M.S. degree in 1966 from Tokyo Institute of Technology. In 1966 he joined in the Electro Technical Laboratories, Tsukuba. He received the Dr.Eng. degree in 1980. He joined in Tokyo Institute of Technology in 1983. He has been engaged in artificial intelligence and natural language processing research.

(Received October 25, 1999 )
(Revised December 6, 1999 )
(Accepted January 14, 2000 )

*Natural Language Engineering* **6** (1): 47–61. Printed in the United Kingdom



# *The LiLFeS Abstract Machine and its Evaluation with the LinGO Grammar*

YUSUKE MIYAO

*Department of Information Science, Graduate School of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033 Japan*
*e-mail:* yusuke@is.s.u-tokyo.ac.jp

TAKAKI MAKINO

*Department of Information Science, Graduate School of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033 Japan*
*e-mail:* mak@is.s.u-tokyo.ac.jp

KENTARO TORISAWA

*Department of Information Science, Graduate School of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033 Japan*
*Information and Human Behavior, PRESTO, Japan Science and Technology Corporation, 4-1-8 Kawaguchi Hon-cho, Kawaguchi-shi, Saitama 332-0012 Japan*
*e-mail:* torisawa@is.s.u-tokyo.ac.jp

JUN-ICHI TSUJII

*Department of Information Science, Graduate School of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033 Japan*
*CCL, UMIST, P.O.Box 88, Manchester, M60 1QD, England*
*e-mail:* tsujii@is.s.u-tokyo.ac.jp

(*Received 15 October 1999; revised 18 February 2000*)

**Abstract**

This article evaluates the efficiency of the LiLFeS abstract machine by performing parsing tasks with the LinGO English resource grammar. The instruction set of the abstract machine is optimized for efficient processing of definite clause programs and typed feature structures. LiLFeS also supports various tools required for efficient parsing (e.g. efficient copying, a built-in CFG parser) and the constructions of standard Prolog (e.g. cut, assertions, negation as failure). Several parsers and large-scale grammars, including the LinGO grammar, have been implemented in or ported to LiLFeS. Precise empirical results with the LinGO grammar are provided to allow comparison with other systems. The experimental results demonstrate the efficiency of the LiLFeS abstract machine.

## 1 Introduction

Efficient processing of feature structures is essential for efficient parsing with HPSG-based grammars, and also for practical applications that process real-world linguistic resources by using feature structures. While optimization methods specific to

HPSG parsing have proven to drastically increase parsing speed (Kiefer, Krieger, Carroll, & Malouf, 1999; Oepen & Carroll, this volume; Torisawa, Nishida, Miyao, and Tsujii, this volume), efficient feature structure processing is still required not only for efficient parsing but also to process parse results in various applications.

*The LiLFeS system* (Makino, Torisawa, & Tsujii, 1997) is a solution for providing a programming environment with efficient processing of feature structures. The LiLFeS language is an extension of Prolog for expressing typed feature structures instead of first order terms. Large-scale systems, not limited to HPSG grammars, can be easily developed on LiLFeS because feature structure descriptions and definite clause programs are consistently integrated. Several large-scale applications have already been developed on this system. Examples include wide-coverage Japanese and English grammars (Mitsuishi, Torisawa, & Tsujii, 1998; Tateisi, Torisawa, Miyao, & Tsujii, 1998), and a statistical disambiguation module for the Japanese grammar (Kanayama, Torisawa, Mitsuishi, & Tsujii, 1999).

The system's core engine is an abstract machine that can process feature structures and execute definite clause programs. In other abstract machine approaches, feature structure processing is separated from execution of programs such as parsers. On the other hand, the LiLFeS abstract machine increases processing speed by seamlessly processing feature structures and executing definite clause programs; it directly executes instructions compiled from the LiLFeS language. This approach also enables efficient low-level manipulation of feature structures, such as block copying and block equivalence checking.

The goal of this article is to evaluate the performance of the LiLFeS abstract machine and to explain how the abstract machine provides efficient processing of feature structures. The performance is evaluated with precise empirical results with the LinGO English resource grammar (Flickinger, this volume) to allow the comparison with other systems in this volume. The LinGO grammar is successfully translated to LiLFeS with the help of the LKB system (Copestake, 1992). This article also describes this translation process of the LinGO grammar and discusses the requirements for running the LinGO grammar on the LiLFeS system. Note that this article does not aim at describing the LiLFeS implementation in detail, which is exhaustively reported in other literature (Makino et al., 1997; Makino, Yoshida, Torisawa, & Tsujii, 1998; Makino, 1999).

Section 2 describes the design of LiLFeS and discusses the translation of the LinGO grammar into the LiLFeS language. Section 3 describes the advantages of the LiLFeS abstract machine architecture. Section 4 reports empirical results on the parsing performance of the abstract machine with the translated LinGO grammar. Section 5 introduces ongoing work aimed at further improvement in feature structure processing.

## 2 LiLFeS as a programming language

This section describes the LiLFeS language and the translation of the LinGO English resource grammar written in the $\mathcal{TDL}$ syntax (Uszkoreit et al., 1994). Since the formal definition of the LiLFeS language appears in another paper (Makino, 1999),

```
head <- [bot].
valence <- [bot] + [SUBJ\list, COMPS\list, SPR\list].
category <- [bot] + [HEAD\head, VAL\valence].
local <- [bot] + [CAT\category, CONT\bot].
synsem <- [bot] + [LOCAL\local, NONLOCAL\bot].
sign <- [bot] + [PHON\list, SYNSEM\synsem].
word <- [sign].
phrase <- [sign] + [HEAD_DTR\sign, NONHEAD_DTR\sign].
id_schema <- [pred].
head_feature_principle <- [pred].
lexical_entry <- [pred].
parse <- [pred].
parse_ <- [pred].
```

*Type definitions*

```
id_schema("head subject schema", $LEFT, $RIGHT, $HEAD, $NONHEAD, $MOTHER) :-
    $LEFT = $NONHEAD,
    $RIGHT = $HEAD,
    $MOTHER = (HEAD_DTR\($HEAD & SYNSEM\LOCAL\CAT\VAL\SUBJ\[$SYNSEM]) &
               NONHEAD_DTR\($NONHEAD & SYNSEM\$SYNSEM)).
head_feature_principle($HEAD_DTR, $MOTHER) :-
    $HEAD_DTR = SYNSEM\LOCAL\CAT\HEAD\$HEAD,
    $MOTHER = SYNSEM\LOCAL\CAT\HEAD\$HEAD.
parse_([$WORD|$TAIL], $TAIL, _, $LEXICON) :-
    lexical_entry($WORD, $LEXICON).
parse_($SENTENCE, $TAIL, [_|$LENGTH], $MOTHER) :-
    parse_($SENTENCE, $MID, $LENGTH, $LEFT),
    parse_($MID, $TAIL, $LENGTH, $RIGHT),
    id_schema($NAME, $LEFT, $RIGHT, $HEAD, $NONHEAD, $MOTHER),
    head_feature_principle($HEAD, $MOTHER).
parse($SENTENCE, $SIGN) :- parse_($SENTENCE, [], $SENTENCE, $SIGN).
```

*Definite clause programs*

Fig. 1. A sample program written in the LiLFeS language

here we mainly discuss the differences between the $\mathcal{TDL}$ and the LiLFeS languages. The problems and their solutions in the translation process are also discussed in this section.

### *2.1 The LiLFeS language*

The LiLFeS language is a programming language to write definite clause programs with typed feature structures. It is similar to Prolog and has various expressions for both processing feature structures and describing procedures. Since typed feature structures can be used like first order terms in Prolog, the LiLFeS language can describe various kinds of application programs based on feature structures. Examples include HPSG parsers, HPSG-based grammars, and compilers from HPSG to CFG. Furthermore, other natural language processing systems can be easily developed because feature structure processing can be directly written in the LiLFeS language.

Figure 1 shows a sample LiLFeS program, which is a very simple parser and grammar. The LiLFeS language makes a clear distinction between type definitions (the upper section in Figure 1) and definite clause programs with feature structures (the lower section in Figure 1). A type (e.g. `phrase`) is defined by specifying supertypes (e.g. `sign`) and features (e.g. `HEAD_DTR\`, `NONHEAD_DTR\`) with their appropriate types (e.g. `sign`). After defining the types, we can use an instance of a feature structure as a first order term in the Prolog syntax. For example, in the predicate `head_feature_principle`, the `$HEAD_DTR` and `$MOTHER` are variables and supposed to be the structure of a sign defined in the type definition section. This predicate is called in the predicate `parse_` in order to apply the *Head Feature Principle* (Pollard & Sag, 1994). Having the same name variable indicates they are structure-shared.

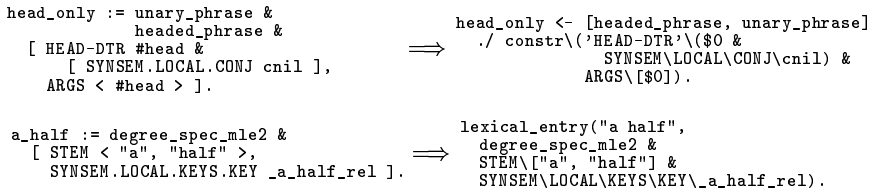



Fig. 2. The translation of type definitions (above) and lexical entries (below) from the $\mathcal{TDL}$ syntax to the LiLFeS syntax

In `head_feature_principle`, the `HEAD` features of the `$HEAD_DTR` and `$MOTHER` are shared because they are indicated with the same variable `$HEAD`.

The differences between the LiLFeS language and the $\mathcal{TDL}$ are summarized in the following.

- The LiLFeS language expresses definite clause programs with typed feature structures, while the $\mathcal{TDL}$ is a typed feature structure description language.
- The LiLFeS language makes a clear distinction between type definitions and instances of typed feature structures, while the $\mathcal{TDL}$ does not distinguish them.

The former says that the description of typed feature structures in the LiLFeS language corresponds to the $\mathcal{TDL}$. Actually, the feature structure description of the LiLFeS language is a syntactic variant of the $\mathcal{TDL}$. Hence, the translation of typed feature structures from the $\mathcal{TDL}$ to the LiLFeS language is simply a syntactic conversion.

On the other hand, the latter indicates that type definitions require a more complex translation process. The type definition in the LiLFeS language follows *totally well-typed feature structures* (Carpenter, 1992). That is, LiLFeS presupposes that type hierarchies are well formed and all types must be defined with their appropriate features. In contrast, the $\mathcal{TDL}$ can describe type definitions which do not follow totally well-typedness, and in addition the LinGO grammar written in the $\mathcal{TDL}$ is based on a slightly different type system, as specified in the Appendix (Copestake, this volume). There are two essential differences which can be summed up as follows.

- Type hierarchies in LiLFeS must be a bounded complete partial order (Carpenter, 1992).
- Each type in LiLFeS is associated with its appropriate features, each of which is associated with an appropriate type.

The type system assumed in the LinGO grammar violates these conditions and therefore we cannot translate type definitions in the LinGO grammar straightforwardly. These problems are discussed next.

### *2.2 Translation of the LinGO grammar*

An HPSG-based grammar is described by using *lexical entries* and *grammar rules*, both of which are denoted by *typed feature structures*. As a result, we have to translate type definitions, lexical entries, and grammar rules. Figure 2 is an example of translating type definitions and lexical entries from the $\mathcal{TDL}$ syntax into the LiLFeS syntax. Note that the LiLFeS language has clear distinction between the type definitions and the feature structure descriptions, while both are described with the same syntax in the $\mathcal{TDL}$.

Translation of lexical entries (and grammar rules) is simple because they are instances of feature structures and the typed feature structure description of the LiLFeS language is directly converted to the $\mathcal{TDL}$ as discussed above. In the lower section of Figure 2, a definition of a lexical entry is converted to the second argument of the predicate `lexical_entry`. This is simply because our parsers on LiLFeS presuppose that lexical entries are provided by the arguments of the predicate `lexical_entry`. Grammar rules are translated in the same way, and they are supposed to be provided by the predicate `id_schema`.

Problems come out when we turn to the translation of type definitions. As described in Section 2.1, the LiLFeS language follows *a totally well-typed feature structure* as defined in Carpenter (1992). Reinterpreting the differences between LiLFeS and the LinGO grammar, the type system in LiLFeS must satisfy the following conditions.

1. Only one least upper bound is allowed for each pair of consistent types
2. An appropriate value for a feature must be a type, not a feature structure

Note that *a greatest lower bound* in Copestake (this volume) corresponds to *a least upper bound* here and in Carpenter (1992). This article follows the definitions in Carpenter (1992).

Condition 1 comes from the condition that the proper type hierarchy is a finite bounded complete partial order, while the type hierarchy in the LinGO grammar does not satisfy this requirement as described in Copestake (this volume). To comply Condition 1, required least-upper-bound types are automatically computed by the LKB system and output for the LiLFeS version of the LinGO grammar (Copestake, this volume). The translated grammar includes 1118 new types for this condition.

Condition 2 is violated by some of the types in the LinGO grammar. For example, `head_only` in Figure 2 is associated with a feature structure. This feature structure is *a type constraint* (Copestake, this volume), which must follow the condition that if a feature structure $F$ is associated with a type $\tau$, a feature structure $F'$ whose root is of the type $\tau$ must be unified with $F$. Apparently, the type constraints cannot be translated to feature appropriateness in the definition of totally well-typed feature structures.

This problem was solved by an extended feature of LiLFeS, which we call *complex constraints*. The extended type definition syntax for complex constraints is shown in Figure 3. The part following `./` defines complex constraints associated with the type *Newtype* as follows.

*Newtype* `<-` `[` *supertypes*... `]` `+` `[` *features*... `]`
`./ constr\` *F* `& pred\` *Q*
*F*: A feature structure constraint
*Q*: A definite clause constraint

Fig. 3. A type definition syntax for defining complex constraints

**Feature structure constraints** $F$ is unified with a feature structure with the associated type.

**Definite clause constraints** $Q$ is executed when the associated type is generated.

The feature structure constraints correspond to the type constraints in Copestake (this volume).

The complex constraint solver is triggered when a type of a node in a feature structure is *changed* into the constrained type or its subtype. It is thus guaranteed that the same constraint will not be triggered again on the same node. The triggered constraint works as if a clause P($F$):–$Q$. is called with $F$ assigned to the changed node. If two or more constraints are triggered on a node at the same time, the order of execution is determined by the types associated to the constraints; a constraint of a certain type is executed earlier than constraints of its subtypes. As for constraints of two types without subsumption relation and constraints on different nodes, their execution order is arbitrary.

By using this extension, type constraints are successfully translated to the LiLFeS language. In Figure 2, a feature structure associated with `head_only` written in the $\mathcal{TDL}$ is converted to a feature structure in the LiLFeS language and specified following `constr\`. The translated LinGO grammar includes 721 types which are converted by using complex constraints.

The LinGO grammar was successfully ported to LiLFeS in a short period with the help of the LKB system and the extended feature of the LiLFeS language. The next section describes the mechanism for providing an efficient programming environment for HPSG-based systems including the translated LinGO grammar.

## 3 The LiLFeS abstract machine

The LiLFeS abstract machine is the core portion of the LiLFeS system, as shown in Figure 4. Since the LiLFeS language is a base system for developing natural language processing systems, the abstract machine should have i) efficient feature structure processing and ii) the ability to execute application programs. To provide these requirements, our research has focused on:

- Efficient implementation of feature structure operations frequently used in parsing tasks:
  - — Unification of typed feature structures
  - — Copying and equivalence checking
- Instructions and data structures required for application programs

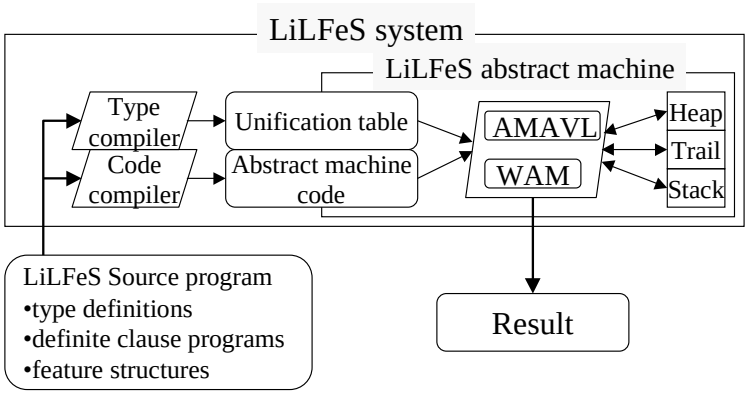



Fig. 4. The architecture of the LiLFeS system

— Execution of definite clause programs such as cut, findall, and negation as failure
— Standard data types such as strings, integers, floating point values, and arrays

For the efficient processing of typed feature structures, we used the *Abstract Machine for Attribute-Value Logics* (AMAVL), the architecture proposed by Carpenter & Qu (1995). By using AMAVL, the LiLFeS abstract machine benefits from i) compact memory representation of typed feature structures and ii) efficient unification by pre-compiling feature structures. In addition, the architecture has been extended for standard data types and arrays. To execute definite clause programs, we adopted *Warren's Abstract Machine* (WAM) (Aït-Kaci, 1991). WAM supports efficient backtracking, as well as cut, negation as failure, and predicate indexing. These two abstract machines are integrated into the LiLFeS abstract machine in an efficient way (Makino et al., 1997), as the data structures and abstract machine instructions of the two abstract machines are unified and re-designed.

The LiLFeS abstract machine has been extended further with efficient built-in subroutines such as efficient copying and equivalence checking of typed feature structures with the concept of *normalization*. Since these subroutines are frequently used in parsing, the increase in speed in these subroutines significantly improves parsing efficiency. In addition, a constraint solving mechanism associated with a type is implemented to handle the complex constraints introduced in Section 2.2.

Consequently, the LiLFeS abstract machine is an integration of the feature structure processing capabilities of AMAVL and the definite clause program execution function of WAM, with extensions required for natural language processing such as the efficient built-in subroutines and the complex constraints. Since the architecture of the abstract machines has already been published (Carpenter & Qu, 1995; Aït-Kaci, 1991; Makino et al., 1997), this section describes the extensions of the LiLFeS abstract machine.

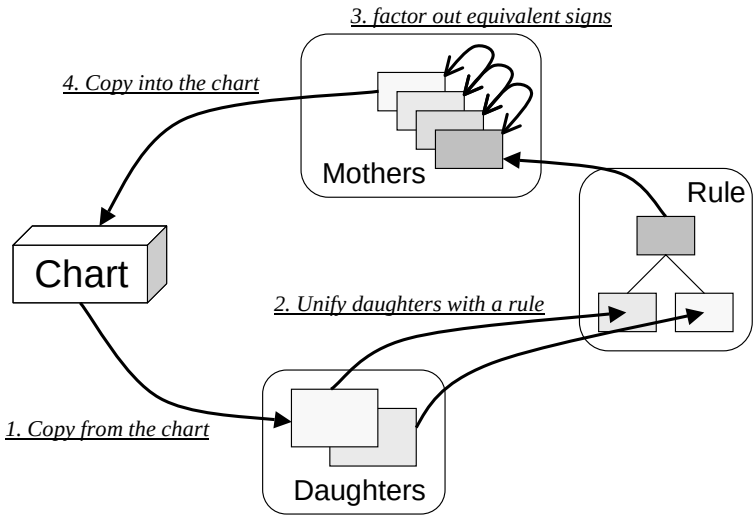



Fig. 5. Operations in a chart parser

### *3.1 Efficient built-in subroutines*

Efficient copying and equivalence checking are especially useful in HPSG parsing because these operations consume a significant proportion of the total parsing time. Figure 5 illustrates operations in a chart parser. First an edge is retrieved from the chart and copied into a working area (1. copy from the chart). A rule is then applied to the copied feature structures (2. Unify daughters with a rule). If it succeeds, the result is compared to previously stored results for factoring out equivalent edges (3. factor out equivalence signs). Finally, the result is stored in the chart (4. copy into the chart). This chart parser is slightly different from those in other systems such as the LKB system. In our system, daughters are copied from the chart and the result is again copied into the chart. By copying the daughters to the working area in each rule application, the rule is applied with purely destructive unification without saving any trails for backtracking. This is faster than unification with backtracking.

However, this parsing algorithm requires extensive copying. The cost cannot be ignored if a naïve copying algorithm is used by traversing a feature structure. The equivalence checking for the edge factoring is also expensive because it requires simultaneous traversing on two typed feature structures and occurrence checking to avoid infinite loops on cyclic feature structures. The factoring is therefore not widely used, although it can reduce the number of rule applications and edges stored in the chart. These problems are solved when optimized built-in subroutines for efficient parsing are provided on the compact memory representation inherited from AMAVL.

To avoid inefficiency in copying, the LiLFeS abstract machine supports *block copying*, which does not need to traverse a feature structure (Makino et al., 1997; Brown & Manandhar, 1998, 2000). As shown in Figure 6, when the LiLFeS abstract machine copies a typed feature structure, it is first *normalized*. That is, the copied

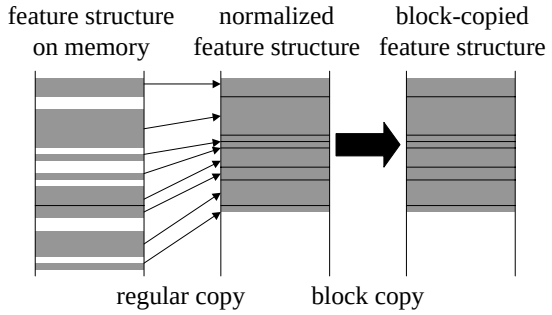



Fig. 6. Block copying

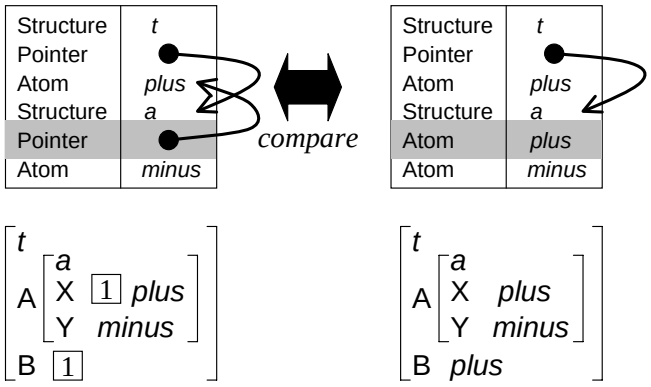



Fig. 7. Block equivalence checking

structure is put on a contiguous block in memory and sorted into a normal form. When copying this kind of normalized feature structure again, we can use block copying; copying the memory block in a linear way.

The factoring also benefits from normalization. *Block equivalence checking*, equivalence checking between two normalized feature structures, is more efficient than regular equivalence checking (see Figure 7). The block equivalence checking regards two normalized typed feature structures as two memory blocks to compare them in the same way as the block copying. Note that the two memory blocks are exactly the same only when they are equivalent feature structures. By using this routine within the factoring operation, recursively traversing a feature structure is not necessary, so factoring costs are drastically reduced.

It should be noted that the LiLFeS abstract machine requires application programmers to call a memory freeing predicate explicitly. This is because the automatic garbage collection involves processing overhead, and also because we can take full advantages of the built-in subroutines. It is hard to implement those subroutines with low overhead on systems that depend on other high-level programming language systems, such as Lisp and Prolog, since their own 'advanced' memory management interferes such subroutines. The parsers we have implemented on LiLFeS

free memory whenever they finish parsing a sentence, so that the abstract machine is able to free all memory blocks that were used during the parsing.

### *3.2 Complex constraint solver*

*The complex constraint* introduced in Section 2.2 is a constraint associated with a type $\tau$ and applied to every feature structure whose root node is of type $\tau$ or any of $\tau$'s subtypes. Any feature structures and definite clause programs can be used to express the constraint, which may include backtracking, cut, negation-as-failure, assertion, and any other operations.

The implementation of complex constraints is tricky; a naïve implementation, that calls up definite clause programs at the moment that a constraint type is generated, cannot be used since this may be unsound as it may call up programs on a partially unified structure. As a result, we chose the following method for implementation:

1. If a constrained type is generated during unification, a continuation frame, that points to the constraint solving routine, is pushed onto the stack.
2. When the unification is completed, each continuation frame is popped from the stack and its corresponding constraint is solved.
3. When all the constraints have been solved, the control is returned to the original routine.

The advantage of this method is that it does not require special handle routines to backtracking. Even when backtracking requires previous continuation frames to be restored on the stack, this is straightforward in this implementation because LiLFeS, as well as WAM, are able to preserve the content of popped portions of the stack for later recovery.

## 4 Performance evaluation

This section evaluates the efficiency of the LiLFeS abstract machine by performing parsing tasks with the LinGO grammar. The following six parsers were compared:

**LKB (vanilla)** The LKB system (Copestake, 1992, 1999) with all filtering methods disabled, simulating the LiLFeS naïve parsing strategy.

**LKB (release)** The parser in the October 1999 release version of the LKB system with filtering methods (Malouf, Carroll, and Copestake, this volume).

**LKB (current)** The parser in the current LKB development version with filtering and parsing strategy optimizations (Oepen & Carroll, this volume).

**Naïve** The CKY-style parser on LiLFeS with no filtering methods.

**TNT** The TNT parser on LiLFeS with CFG filtering (Torisawa et al., this volume).

**TNT*** The TNT parser with a threshold number of edges in the first (CFG) parsing phase (see Torisawa et al., this volume for details).

The following experiments are performed on three test suites described in Introduction in this volume, namely, '*csli*', '*aged*', and '*blend*'. Sentences that required

| test set | parser | etasks | stasks | tgc (s) | total (s) | etasks/s |
|---|---|---|---|---|---|---|
| *'csli'* | **LKB (vanilla)** | 36953 | 4308 | 0·18 | 3·98 | 9691 |
| | **LKB (release)** | 658 | 556 | 0·02 | 0·39 | 1760 |
| | **LKB (current)** | 359 | 183 | 0·00 | 0·23 | 1517 |
| | **Naïve** | 28002 | 3788 | – | 0·68 | 41094 |
| | **TNT** | 276 | 145 | – | 0·12 | 2300 |
| *'aged'* | **LKB (vanilla)** | 107219 | 13164 | 0·12 | 11·87 | 9060 |
| | **LKB (release)** | 1842 | 1604 | 0·07 | 1·13 | 1725 |
| | **LKB (current)** | 866 | 452 | 0·00 | 0·61 | 1409 |
| | **Naïve** | 68188 | 10002 | – | 1·72 | 39672 |
| | **TNT** | 668 | 379 | – | 0·31 | 2155 |
| *'blend'* | **LKB (vanilla)** | 703251 | 84755 | 2·97 | 80·99 | 9009 |
| | **LKB (release)** | 10493 | 9124 | 0·25 | 5·94 | 1839 |
| | **LKB (current)** | 3738 | 1685 | 0·12 | 3·10 | 1252 |
| | **Naïve** | 346772 | 70535 | – | 14·71 | 23574 |
| | **TNT** | 11261 | 4938 | – | 3·67 | 3068 |
| | **TNT*** | 9753 | 4308 | – | 1·90 | 5133 |

Table 1. *Parsing performance with the LinGO grammar*

more than 20,000 edges to parse were removed. The profiling data were collected by using [incr tsdb()] (Oepen & Flickinger, 1998; Oepen & Carroll, this volume) except for the TNT parser. The use of [incr tsdb()] also guarantees that the derivations obtained on LiLFeS are the same as the results of the LKB system. All the experiments were conducted on a 336 megahertz Sun UltraSparc with six gigabytes of memory.

We should note that *LKB (vanilla)* and *Naïve* are based on almost the same parsing algorithm, that is, they require almost the same number of rule applications (i.e. *etasks*). However, we must also mention that the number of tasks should be larger with *LKB (vanilla)* than with *Naïve*, because the grammars they use are slightly different. All lexical entries were precompiled (i.e. all lexical rules are already applied) and no lexical rules are included in the LiLFeS version of the LinGO grammar. The LiLFeS version has thirty seven grammar rules, while the original LinGO grammar has thirty seven grammar rules plus twenty seven lexical rules. *LKB* can filter out lexical rule applications to phrasal signs and does not suffer from the overhead of the application of the lexical rules.

### *4.1 Empirical results with the LinGO grammar*

Table 1 shows the experimental results of parsing with the LinGO grammar and compares five parsers. The meanings of the values in each column are described in Oepen & Carroll (this volume). The last column shows the number of *etasks* per second, which provides an approximate measure of feature structure processing (mostly unification speed).

| parser | preprocessing | filtering | parsing | total |
|---|---|---|---|---|
| **Naïve** | 56 | 0 | 662 | 718 |
| **TNT** | 23 | 43 | 38 | 108 |

Table 2. *Time for preprocessing, filtering, and parsing (ms)*

Comparing the last column of *LKB (vanilla)* and *Naïve*, we can clearly see that the LiLFeS abstract machine processes feature structures very efficiently. The LiLFeS abstract machine can process (unify) typed feature structures around four times faster than the LKB system. While the number of *etasks* and *stasks* is significantly smaller in *LKB* due to the filtering methods, *Naïve* still offers a competitive parsing speed without using any filtering methods.

Comparing *LKB* and *TNT*, we notice that efficient feature structure processing is still essential with parsers based on advanced algorithms. When they performed almost the same number of *etasks* and *stasks*, *TNT* achieved faster parsing speed thanks to the efficiency of the LiLFeS abstract machine.

*Naïve* vs. *TNT* (and *LKB (vanilla)* vs. *LKB*) shows the effectiveness of the filtering method. The number of *etasks* was significantly reduced in *TNT* compared with the others, and the parsing speed was drastically increased. The number of unifications per second (*etasks/s*) is smaller than that in *Naïve*, because some percentage of the parsing time is consumed in filtering out unnecessary tasks and a successful unification tends to take more time than a failed unification. Note that the *stasks* value is also significantly reduced in *TNT*. This is because CFG filtering can approximate not only local constraints but also global constraints (Torisawa et al., this volume).

Table 2 shows the time consumed for preprocessing, filtering, and parsing in *Naïve* and *TNT* with the '*csli*' test suite. Since sentences in the test corpus are short, the preprocessing requires significant time in the overall parsing time. We found that most of the preprocessing time is consumed at fetching lexical entries. The cost for looking up lexical entries should be optimized in future research, although it might be less significant with longer sentences.

### *4.2 Efficiency of unification and built-in subroutines*

Table 3 shows the time for the unification and other operations (copying and equivalence checking) in parsing the '*csli*' test suite. The total parsing time is slightly larger than the results in Table 1 because of the overhead for measuring the time. Unification accounts for most of the parsing time. When the efficient built-in subroutines are used (i.e. block copying and block equivalence checking), their computational cost is negligible as can be seen on the first row. The second row provides processing times without factoring, (i.e. without any equivalence checking). In this case, the efficiency of the block equivalence checking becomes evident as the processing time for the first two rows is virtually the same. This is reinformed by the

| | parser | unification | copying & equiv. checking | total |
|---|---|---|---|---|
| **Naïve** | (with subroutines) | 610 | 46 | 775 |
| | (no factoring) | 615 | 45 | 770 |
| | (without subroutines) | 625 | 196 | 877 |
| **TNT** | (with subroutines) | 30 | 17 | 123 |

Table 3. *Time for unification, copying, and equivalence checking (ms)*

poor performance without the efficient built-in subroutines (the third row). The fourth row shows the time consumption for the unification and built-in subroutines in the TNT parser. It shows that the efficiency of the built-in subroutines is more important in parsers based on an advanced parsing algorithm.

## 5 Ongoing work

We are researching several more methods to further increase the speed of feature structure processing. The following methods have already been evaluated and shown their effectiveness at an experimental level. They will be integrated within the current parsers and should contribute to their improvement.

**The LiLFeS native code compiler** (Makino, 1999) generates native machine code for a Pentium CPU from a LiLFeS source code. An abstract machine instruction is decomposed and directly compiled to several CPU-level instructions. The abstract machine code is optimized in compile-time by referring to the result of dataflow analysis with abstract interpretation. In the current state the compiled code is more than two times faster in parsing sentences than the abstract machine emulator. Unfortunately the parsing speed with the LinGO grammar cannot be measured because complex constraints have not been implemented yet.

**Feature structure packing** (Miyao, 1999) aims at processing feature structures with disjunctions efficiently. A set of non-disjunctive feature structures is automatically converted into a more compact data structure, *a packed feature structure*, by collapsing equivalent parts in the input feature structures. Since any unification operation is performed on the collapsed parts only once, unification speed is significantly improved for feature structures with many disjunctions. Even when the factoring operation is not effective, such as in parsing with the LinGO grammar, the packing method should improve the performance of the system significantly. Preliminary experiments show that the unification speed improved by a factor of 6.4 to 8.4.

## 6 Conclusion and future work

The LiLFeS system processes typed feature structures efficiently by taking an abstract machine approach. The efficiency of feature structure processing in the LiL-

FeS abstract machine is apparent in the experimental results done with the LinGO grammar ported to LiLFeS. Along with the efficient parsing systems, we are investigating several applications with HPSG-based grammars. Some research, such as statistical parsing (Kanayama et al., 1999) and robust parsing (Steiner & Tsujii, 1999a, 1999b) to support these applications is being conducted. Furthermore, the LiLFeS system is used not only for HPSG-based systems but also for other formalisms, such as an efficient TAG parser (Yoshida, Ninomiya, Torisawa, Makino, & Tsujii, 1999).

## Acknowledgments

This article could not have been completed without the help of Dr. Ann Copestake. She helped with the translation of the LinGO grammar by using the LKB system. We are also indebted to Mr. Stephan Oepen for letting us use his profiling system [incr tsdb()] and his detailed discussion on the framework. We would like to thank Mr. Takashi Ninomiya for making a number of helpful suggestions and integrating [incr tsdb()] with the LiLFeS system. Finally, we thank the anonymous reviewers for their valuable comments on this manuscript.

## References

Aït-Kaci, H. (1991). *Warren's Abstract Machine: A tutorial reconstruction.* Cambridge, MA: MIT Press.

Brown, J. C., & Manandhar, S. (1998). *An abstract machine for fast parsing of typed feature structure grammars* (Tech. Rep.). Computer Science Department of University of the Saarlandes.

Brown, J. C., & Manandhar, S. (2000). An abstract machine for fast parsing of typed feature structure grammars: including experimental results. In *Proceedings of Future Generation Computer Systems.*

Carpenter, B. (1992). *The logic of typed feature structures.* Cambridge, UK: Cambridge University Press.

Carpenter, B., & Qu, Y. (1995). An abstract machine for attribute-value logics. In *Proceedings of the 4th International Workshop on Parsing Technologies.* Prague, Czech Republik.

Copestake, A. (1992). The ACQUILEX LKB. Representation issues in semi-automatic acquisition of large lexicons. In *Proceedings of the 3rd ACL Conference on Applied Natural Language Processing* (pp. 88–96). Trento, Italy.

Copestake, A. (1999). *The (new) LKB system.* (CSLI, Stanford University: `http://www-csli.stanford.edu/~aac/doc5-2.pdf`)

Kanayama, H., Torisawa, K., Mitsuishi, Y., & Tsujii, J.-I. (1999). Statistical dependency analysis with an HPSG-based Japanese grammar. In *Proceedings of the Natural Language Processing Pacific Rim Symposium* (pp. 138–143). Beijing, China.

Kiefer, B., Krieger, H.-U., Carroll, J., & Malouf, R. (1999). A bag of useful techniques for efficient and robust parsing. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics* (pp. 473–480). College Park, MD.

Makino, T. (1999). *A native-code compiler for a unification-based programming language with typed feature structures.* Unpublished master's thesis, University of Tokyo.

Makino, T., Torisawa, K., & Tsujii, J.-I. (1997). LiLFeS — practical programming language for typed feature structures. In *Proceedings of the Natural Language Processing Pacific Rim Symposium.*

Makino, T., Yoshida, M., Torisawa, K., & Tsujii, J. ichi. (1998). LiLFeS — towards a practical HPSG parser. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (pp. 807−11). Montreal, Canada.

Mitsuishi, Y., Torisawa, K., & Tsujii, J.-I. (1998). HPSG-style underspecified Japanese grammar with wide coverage. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (pp. 876−80). Montreal, Canada.

Miyao, Y. (1999). Packing of feature structures for efficient unification of disjunctive feature structures. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics* (pp. 579−84). College Park, MD.

Oepen, S., & Flickinger, D. P. (1998). Towards systematic grammar profiling. Test suite technology ten years after. *Journal of Computer Speech and Language, 12 (4) (Special Issue on Evaluation)*, 411−436.

Pollard, C., & Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar.* Chicago, IL and Stanford, CA: The Univeristy of Chicago Press and CSLI Publications.

Steiner, R., & Tsujii, J.-I. (1999a). M-rules: Adaptable rules for robustness in unification-based systems. In *Proceedings of the Natural Language Processing Pacific Rim Symposium* (pp. 126−131). Beijing, China.

Steiner, R., & Tsujii, J.-I. (1999b). Robustness in HPSG via extended ID schemata. In *Proceedings of the Natural Language Processing Pacific Rim Symposium* (pp. 108−113). Beijing, China.

Tateisi, Y., Torisawa, K., Miyao, Y., & Tsujii, J.-I. (1998). Translating the XTAG English grammar to HPSG. In *Proceedings of the TAG+4 workshop* (pp. 172−175). Philadelphia, PA.

Uszkoreit, H., Backofen, R., Busemann, S., Diagne, A. K., Hinkelman, E. A., Kasper, W., Kiefer, B., Krieger, H.-U., Netter, K., Neumann, G., Oepen, S., & Spackman, S. P. (1994). DISCO — an HPSG-based NLP system and its application for appointment scheduling. In *Proceedings of the 15th International Conference on Computational Linguistics.* Kyoto, Japan.

Yoshida, M., Ninomiya, T., Torisawa, K., Makino, T., & Tsujii, J.-I. (1999). Efficient FB-LTAG parser and its parallelization. In *Proceedings of the conference of the Pacific Association for Computaional Linguistics.* Waterloo, Canada.

# 用言と直前の格要素の組を単位とする格フレームの自動獲得

河原大輔　黒橋禎夫
京都大学大学院 情報学研究科
〒 606-8501 京都市左京区吉田本町
{kawahara,kuro}@pine.kuee.kyoto-u.ac.jp

**あらまし**

本稿では、生コーパスから格フレームを自動的に獲得する手法を提案する。まず、生コーパスを解析し、解析結果から確信度の高い係り受けを収集する。次に、用言の意味的曖昧性に対処するために、用言と直前の格要素の組を単位として、生コーパスから用例を収集し、それらのクラスタリングを行う。また、得られた格フレームを用いて格解析を行い、その結果を示す。

# Case Frame Construction by Coupling the Predicate and its Adjacent Case Component

Daisuke Kawahara　Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501 JAPAN
{kawahara,kuro}@pine.kuee.kyoto-u.ac.jp

**Abstract**

This paper describes a method to construct a case frame dictionary automatically from a raw corpus. First, we parse a corpus and collect reliable examples from the parsed corpus. Secondly, to deal with semantic ambiguity of a predicate, we distinguish examples by a predicate and its adjacent case component and cluster them. We also report on an experimental result of case structure analysis using the constructed dictionary.

# 1　はじめに

日本語には語順の入れ替わり、格要素の省略、表層格の非表示などの問題があり、単純な係り受け解析を行っただけでは文の解析として十分とはいえない。例えば、「ドイツ語も話す先生」という文の場合、係り受け構造を解析しただけでは、「ドイツ語」と「話す」、「先生」と「話す」の関係はわからない。このような問題を解決するためには、用言と格要素の関係、例えば、「話す」のガ格やヲ格にどのような単語がくるかを記述した格フレームが必要である。さらに、このような格フレームは文脈処理 (照応処理、省略処理) においても必須の知識源となる。

格フレーム辞書を構築する方法のひとつは、人手による作成である [3, 6]。しかし、その場合には非常に大きなコストがかかり、カバレージの大きな辞書を作成、保守することは難しい。また、これまでの人手による辞書では、格と同じ振る舞いをする「によって」、「として」などの複合辞や、「〜が〜に人気だ」のように名詞+判定詞の格フレームを取り扱っているものはほとんどない。

そこで、格フレーム辞書をコーパスから自動学習する方法を考える必要がある。しかし、格フレームの学習には膨大なデータが必要となり、現存するタグ付きコーパスはこのような目的からは量的に不十分である。そこで、本論文では、格フレーム辞書をタグ情報が付与されていない大規模コーパス (生コーパス) から自動的に構築する手法を提案する。

格フレーム辞書を生コーパスから学習するためには、まず、生コーパスを構文解析しなければならないが、ここで解析誤りが問題となる。しかし、この問題はある程度確信度が高い係り受けだけを学習に用いることでほぼ対処することができる。むしろ問題となるのは用言の意味の曖昧性である (これはタグ付きコーパスから学習する場合にも問題となる)。つまり、同じ表記の用言でも複数の意味をもち、意味によってとりうる格や体言が違うことがあるので、用言の意味ごとに格フレームを作成することが必要である。本論文では、これに対処するために、

1. 用言とその直前の格要素の組を単位として用例をまとめ、
2. さらに、それらのクラスタリングを行った。

# 2　格フレーム学習の種々の方法

生コーパスからの格フレーム辞書の構築の過程は以下のとおりである (図 1 の点線で囲まれた部分)。

1. コーパスのテキストに対して、KNP[5] を用いて構文解析を行い、その結果から、ある程度信頼できる用言・格要素間の関係を取り出す。ここで取り出すデータを**用例**と呼ぶ。
2. 抽出した関係を用言と直前の格要素の組ごとにまとめる。このようにして作成したデータを**用例パターン**と呼ぶ。
3. シソーラスを用いて、用例パターンのクラスタリングを行う。この結果できたものを**用例格フレーム**と呼び、本研究ではこれが最終的に得られるものである。以下では「雑誌」、「本」、「先」などの格要素になる単語を**格用例**、用例格フレームにおけるある格の格用例の集合、例えば「読む」のひとつめの用例格フレームのヲ格の格用例集合、{「雑誌」, 「本」} を**格用例群**と呼ぶ。

次に、格フレームに関連するさまざまなデータ処理を図 1 に沿って議論する。

まず、図 1 の I の用例をそのまま個別に使うことが考えられるが、この場合データスパースネスが問題になる。例えば、

(1) a. 図書館で本を読む
　　b. 家で新聞を読む

という 2 つの用例がコーパスにあったとしても、「図書館で新聞を読む」という表現が妥当であるかどうかはわからない。

一方、用例を二項関係に分割すると、図 1 の II のような共起データを作ることができる。これは統計パーサによって用いられているデータ形式であり、データスパースネスの問題を回避することができる [1]。しかし、その副作用として用言の意味の曖昧性の問題が生じる。例えば、

(2) a. 彼が天を仰ぐ
　　b. 彼が先生に指示を仰ぐ

この 2 つの用例から「天を 仰ぐ」、「先生に 仰ぐ」、「指示を 仰ぐ」という共起データが得られるが、これらのデータだけでは「先生に天を仰ぐ」のような間違った表現を許すことになる。

また、図 1 の III のように用例を単純にまとめたものも、もっている情報は共起データと同じであ

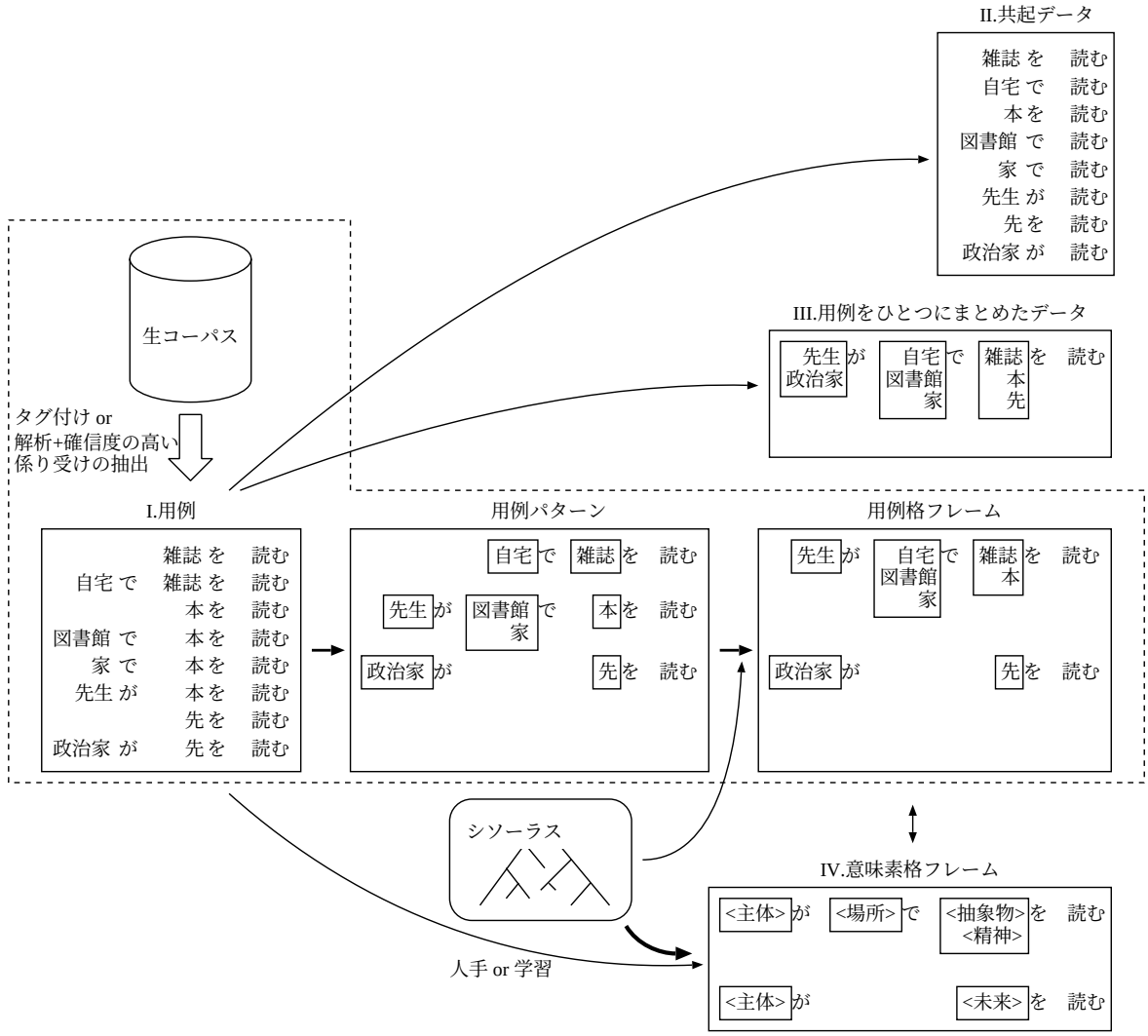



図 1: 格フレームに関連するさまざまなデータ処理

り、やはり用言の意味の曖昧性が問題となる。

本手法は、用言とその直前の格要素を組にして扱うという方法で、意味の曖昧性の問題を解消しつつ、データスパースネスにも対処している。

一方、用例格フレームの格要素を意味素に抽象化することにより、図 1 の IV に示したような**意味素格フレーム**を考えることができる。しかし、これを人手で作成する場合は、1 章で述べたような問題がある。宇津呂らは意味素格フレームをコーパスから学習している [4]。本研究との違いは、学習にタグ付きコーパスを用いていること、また、用言の意味の曖昧性を扱っていないことである。

また、意味素格フレームの作成はシソーラスに深く依存してしまう。一般に、シソーラスにおいて、同義語、類義語を定義するには問題ないが、それを大きな体系にまとめる部分では大きく主観に依存してしまう。意味素格フレームの作成は、そのような大きな体系に左右されるので、シソーラスの不整合な問題の影響を受けることになる。本研究では、シソーラスの比較的安定した部分だけを用いており、シソーラスによる悪影響はほとんど出ない。

## 3 用例の収集

コーパスを構文解析した結果から、図 1 に示したような用例の収集を行う。単語を個別に扱うことにあまり意味がなく、意味が明確な格要素はクラスとして扱う。また、質の高い用例を収集するために、コーパスの解析結果から確信度の高い係り受けを抽出するということを行う。

### 3.1 収集に関する条件

用例を収集するときに、格要素、用言のそれぞれに以下のような条件を設定する。

**格要素の条件**

収集する格要素は、ガ格、ヲ格、ニ格、ト格、デ格、カラ格、ヨリ格、無格とする。また、次のものを新たな格として扱う。

**時間格** ニ格、無格、カラ格、マデ格で、意味素「時間」(3.2 節で述べる) をもっている格要素はまとめてひとつの格にする。これは、格フレームを作るときには、その用言が時間に強く関係しているかどうかが重要であり、そのような格要素は表層格をそのまま扱うよりも、表層格をまとめてひとつの格にしたほうが望ましいからである。

例: 3時に, 来年から

**複合辞** 格と同じように振る舞う複合辞も、ひとつの格として扱う。

例: ～をめぐって, ～によって,
～について, ～として

次のような格要素は収集に用いない。

- 提題助詞をもつ格要素と用言の連体修飾先は、表層格が明示されていないので収集に用いない。

  例: ～を提案している 議員 が～
  その 議員 は～を提案した。

- ニ格、デ格で副詞的に使われる格要素は、係る用言との関係が任意的であるので収集から除外する。

  例: ために, 無条件に, うえで, せいで

- KNP では、「～では」、「～でも」はデ格、「～には」、「～にも」はニ格の格要素として扱われるが、副助詞、あるいは従属節の場合もあるので収集の対象から除外する。

  例: 足の１本 でも 折ってやろうかと思った。
  育成しないこと には 世界で通用しない。

格要素として、文節内の最後の自立語を収集に用いる。格要素が複合名詞の場合には、そのなかで最後の自立語がもっとも意味的に重要であると考えられるからである。

**用言の条件**

収集する用言は動詞、形容詞、名詞+判定詞とする。名詞+判定詞として収集する用言には体言止めの名詞も含む。ただし、以下のような用言は収集に用いない。

- 用言が受身、使役、「～もらう」、「～たい」、「～ほしい」、「～できる」の形であれば、格の交替が起こり、格と格要素の関係が通常の場合と異なるので収集に用いない。

- 「～で」は、判定詞かデ格かの自動判定が難しいので、KNP が判定詞と認識しても、用言として収集に用いない。

  例: 彼は 京都で、試験を受け… (助詞)
  彼が好きな町は 京都で、… (判定詞)

### 3.2 格要素の汎化

個別の単語を扱うことにあまり意味がなく、明確な意味を考えることできる格用例はクラスとしてまとめて扱う。この汎化したクラスを以下のように 3 種類設定した。この場合、格用例として単語のかわりにクラスを記述する。

**時間**

- 品詞細分類が時相名詞
  例: 朝, 春, 来年

- 時間助数辞を含む文節
  例: 年, 月, 日, 時, 分, 秒

- 「前」、「中」、「後」という接尾辞をもち、自立語がシソーラス上の「場所」の意味属性をもたない文節
  例: 会議中, 戦争後, 書く前に

**数量**

- 数詞を含む文節
  例: １, ２, 一, 二, 十, 百

- 数詞と、「つ」、「個」、「人」のような助数辞を含む文節については、「<数量>つ」、「<数量>個」、「<数量>人」のように数量クラスと助数辞のペアにして学習する。

  例: １つ → <数量>つ
  ２個 → <数量>個

**補文**

- 引用節「〜と」、連体修飾+形式名詞 またはそれに準ずる表現 (〜の〜, 〜くらい〜, )

  例: 書くと, 書いたことを, 書くのを, 書くくらいが

例えば、

(3) ３０日に総理大臣がその２人に賞を贈った。

という文からは、

<時間>:時間格　大臣:が

<数量>人:に　賞:を　　贈る

という用例を学習する。

### 3.3　確信度の高い係り受けの抽出

コーパスを構文解析した結果から用例を収集するときに問題となるのは、解析結果に誤りが含まれていることである。そこで、誤りの影響を軽減するために、解析の精度が低い係り受けは捨てて、ある程度確信度が高い係り受けを格フレームの収集に用いる。

KNP では、次のような優先規則によって文節の係り先を決定している。

1. 文中の強い区切りを見つけることによって、係り先の候補の絞り込みを行う (ここで候補がひとつになるなら、係り先をそれに決定する)。
2. 係り先の候補の用言のうち、格要素の係り先にならないことが多い用言を候補から除外する。
3. “読点のない文節はもっとも近い候補に係り、読点のある文節は 2 番目に近い候補に係る” という優先規則に従って、候補の中から係り先を決定する。

用例の収集では、1 は信頼し、2 と 3 は信頼しない (多くの場合正しいが、誤っていることもある) こととする。つまり、1 で候補がひとつになり決定される係り受けは用例の収集に用い、2 や 3 の処理が適用された係り受けは収集に用いない。例えば、

(4) その著者は買いたい 本を たくさん見つけたので、東京へ送った。

この例において、まず、まったく曖昧性なしに取り出すことができる用例は、文末にある「東京へ 送った」だけである。ここでさらに、「〜ので」は KNP によって強い区切りであると認識され、「本を」の係り先の候補は「見つけた」の 1 つしかないので、この用例を取り出す。

次に、2 において、係り先の候補から除外した用言は、場合によっては係り先になる可能性があるので、このときの用例は収集しないことにする。例えば、次の例のように、形容詞のすぐ後に強い用言がある場合、このような形容詞は格要素の係り先になりにくいために、係り先の候補から除外される。

(5) 長女が気づき、家族とともに二人を助けようとしたが火の 回りが 早く救い出せなかった。

この例では、「回りが」は形容詞「早く」に係るのが正解であるが、「早く」は係り先の候補から除外されており解析が誤っている。

また、3 の処理の例を次に示す。

(6) 商工会議所の会頭が、質問に 先頭を切って答えた。

KNP は、「質問に」の係り先の候補として、「切って」、「答えた」の 2 つの可能性を考慮する。この場合、“より近くに係る” という優先規則に従って係り先は「切って」に決定されるが、この解析は誤りである。この例のように、係り先の候補が複数存在すると、係り先に曖昧性があり確信度が低いので、このような用例は収集しない。

## 4　用例格フレームの作成

2 章の例文で示したように、用言の意味の異なる用例をひとつの格フレームとしてまとめて学習すると、誤った表現を許す格フレームを作ってしまう。従って、格パターンの異なる格フレームは別々に学習する必要がある。

用言の意味を決定する重要な格要素は用言の直前にくることが多い。また、用言とその直前の格要素をペアにして考えると、用言の意味の曖昧性はほとんどなくなる。 そこで、用例を、**用言とその直前の格要素の組を単位としてまとめる**という処理を行い、用例パターン (図 1) を作る。用例パターンの用言の直前の格要素を**用例パターンの直前格要素**と呼ぶ。

用例パターンは、ひとつの用言について、直前格要素の数だけ存在している。そのため、次の例のように、意味がほとんど同じパターンまで個別に扱われている。

(7) a.　自宅:で 雑誌:を 読む

b.　先生:が { 図書館, 家 }:で 本:を 読む

そこで、ほとんど意味や格パターンが同じ用例パターンをマージするために、用例パターンのクラス

タリングを行う。以下では、このクラスタリングの詳細について述べる。

### 4.1 用例パターン間の類似度

用例パターンのクラスタリングは、格用例群間の類似度と格の一致度をもとにした用例パターン間の類似度を用いて行う。

まず、日本語語彙大系のシソーラスを利用し、意味属性 $x, y$ 間の類似度 $sim(x, y)$ を次のように定義する。

$$sim(x, y) = \frac{2L}{l_x + l_y}$$

ここで、$l_x$ , $l_y$ は $x$ と $y$ の意味属性のシソーラスの根からの階層の深さを表し、$L$ は $x$ と $y$ の意味属性で一致している階層の深さを表す。類似度 $sim(x, y)$ は 0 から 1 の値をとる。

次に、単語 $e_1, e_2$ 間の類似度 $sim_e(e_1, e_2)$ は、$e_1, e_2$ それぞれの意味属性間の類似度の最大値と考え、

$$sim_e(e_1, e_2) = max_{x \in s_1, y \in s_2}\, sim(x, y)$$

と定義する。$s_1, s_2$ はそれぞれ $e_1, e_2$ の日本語語彙大系における意味属性の集合である (日本語語彙大系では、単語に複数の意味属性が与えられている場合が多い)。

また、格用例群 $E_1, E_2$ 間の類似度 $sim_E$ は、格用例の類似度の和を正規化したもので、

$$sim_E(E_1, E_2) = \frac{\sum_{e_1 \in E_1} \sum_{e_2 \in E_2} \sqrt{|e_1||e_2|}\, sim_e(e_1, e_2)}{\sum_{e_1 \in E_1} \sum_{e_2 \in E_2} \sqrt{|e_1||e_2|}}$$

とする。$|e_1|$ などの絶対値は頻度を表している。ただし、実際には、$sim_e(e_1, e_2)$ の上位から 1/5-best だけを足すことにする。

ここで、用例パターン $F_1$、$F_2$ の格の一致度 $cs$ は、格用例の頻度を重みとした、共通格の割り合いの平方根とし、

$$cs = \sqrt{\frac{\sum_{i=1}^{n} |E_{1cc_i}| + \sum_{i=1}^{n} |E_{2cc_i}|}{\sum_{i=1}^{l} |E_{1c1_i}| + \sum_{i=1}^{m} |E_{2c2_i}|}}$$

と定義する。ただし、用例パターン $F_1$ 中の格を $c1_1, c1_2, \cdots, c1_l$、用例パターン $F_2$ 中の格を $c2_1, c2_2, \cdots, c2_m$、$F_1$、$F_2$ の共通格を $cc_1, cc_2, \cdots, cc_n$ とする。また、$E_{1cc_i}$ は $F_1$ 内の格 $cc_i$ に含まれる格用例群で、他も同様である。

用例パターン $F_1$、$F_2$ 間の類似度 $score$ は、格の一致度 $cs$ と $F_1$、$F_2$ の共通格の格用例群間の類似度の積とし、

$$score = cs \cdot \frac{\sum_{i=1}^{n} \sqrt{w_i}\, sim_E(E_{1cc_i}, E_{2cc_i})}{\sum_{i=1}^{n} \sqrt{w_i}}$$

と定義する。ただし、 $w_i$ は、格用例群間の類似度に対する重みであり、格用例の頻度の平方根をもとにして、

$$w_i = \sum_{e_1 \in E_{1cc_i}} \sum_{e_2 \in E_{2cc_i}} \sqrt{|e_1|\,|e_2|}$$

とする。

### 4.2 用例パターンの意味属性の固定

用例パターン間の類似度は、用例パターンの直前格要素の意味属性が大きく影響する。すると、用例パターンの直前格要素に多義性があるときに問題がある。例えば、「合わせる」の用例パターンのクラスタリングにおいて、用例パターンの組 (手, 顔)[1]が意味属性 <動物（部分）>、(手, 焦点) が意味属性 <論理・意味等> でマージされるときに、<動物（部分）> と <論理・意味等> はまったく類似していない意味属性であるのにもかかわらず、 (手, 顔, 焦点) というマージが起きてしまう。

この問題に対処するために、もっとも類似度が高い用例パターンの組から意味属性を固定する処理、すなわち用例パターンの意味の曖昧性解消を行う。この処理は、用例パターンの直前格要素の意味属性を固定することによって、次のような手順で行う。

1. 類似度が高い用例パターンの組 (p, q) から順に、両方の用例パターンの直前格要素 $\mathrm{n}_p$, $\mathrm{n}_q$ の意味属性を固定する。固定する意味属性は、$\mathrm{n}_p$, $\mathrm{n}_q$ 間の類似度を最大にする意味属性 $\mathrm{s}_p$, $\mathrm{s}_q$ とする。
2. p, q に関係する用例パターンの類似度を再計算する。
3. 閾値 $threshold$ を越える用例パターンの組がなくなるまで、この 2 つの処理を繰り返す。

### 4.3 アルゴリズム

用例パターンのクラスタリングの手順を以下に示す。

[1]ここでは、直前格要素で用例パターンを表している。

1. まず、直前の格要素の出現頻度がある閾値以上あるという条件で足切りを行う。これは、直前の格以外にも格用例がある程度の回数以上出現しているような安定した用例パターンだけを対象にするためである。この閾値は 10 に設定した。
2. あらゆる 2 つ組の用例パターンの類似度を計算し、用例パターンの意味属性を固定する。これらの処理は、4.2 節で述べたように繰り返す。
3. 用例パターン間の類似度が閾値 $threshold$ を越える組について、用例パターンのマージを行う。
4. 頻度の閾値を越えない用例パターン (残りの用例パターン) を作成された用例格フレームにふりわける。用例パターンと用例格フレーム間の類似度を計算し、類似度が閾値 $threshold_r$ を越え、もっとも類似している用例格フレームにマージする。$threshold_r$ は副作用を生まないように、ある程度高い値に設定する。

## 5 必須格の選択

クラスタリングを行った結果得られる用例格フレームについて、格用例の頻度が少ない格は除く。これは、ひとつには構文解析結果の誤りへの対策であり、また頻度の少ない格はその用言と関係が希薄であると考えられるからである。ただし、ガ格についてはすべての用言がとると考え、頻度が少なくても削除せず、逆にガ格の格用例がない場合には、意味属性 <主体> を補うことにした。

頻度の閾値は、現在のところ経験的に $2\sqrt{mf}$ と定めている。ただし、$mf$ はその用言において最も多く出現した格の延べ格用例数である。例えば、$mf$ が 100 のとき、閾値は 20 となり、格用例の頻度が 20 未満の格は捨てられることになる。

## 6 作成した格フレーム辞書

毎日新聞約 7 年分の 360 万文から実際に格フレーム辞書を構築した。クラスタリングの閾値 $threshold$ は 0.65、残りの用例パターンを振り分ける閾値 $threshold_r$ は 0.80 に設定した。これは、格パターンが違ったり、意味が違う格フレームが同じ格フレームにならないという基準で設定したものである。従って、格フレームは基本的にはばらばらで、意味がほとんど同じ格フレームを最小限まとめたものになっている。格フレームの例を表 1 に示す。この表では、<主体>、<場所> の意味属性をもつ格用例を【主体】、【場所】という意味属性でまとめて表示している。

表 1: 構築した格フレームの例 (* はその格が用言の直前の格であることを示す。)

| 用言 | 格 | 用例 |
|---|---|---|
| 買う 1 | ガ格 | 【主体: <数量>人, 乗客 】 |
| | ヲ格* | 株, 円, 土地, もの, ドル, 切符 |
| | デ格 | 【場所: 店, 駅】, <数量>円 |
| 買う 2 | ガ格 | 対応, 厚生, 絵はがき, 蓄財 |
| | ヲ格* | 怒り, ひんしゅく, 失笑, 反感 |
| ： | ： | ： |
| 読む 1 | ガ格 | 【主体: 大学生, 首相, 先生】 |
| | ヲ格* | 本, 記事, 新聞, 小説, 投書 |
| 読む 2 | ガ格 | 【主体: <主体>】 |
| | ヲ格 | 話, <補文>, 意見, 惨状 |
| | デ格* | 新聞, 本, 本紙, 教科書 |
| 読む 3 | ガ格 | 【主体: <主体>】 |
| | ヲ格* | 先 |
| ： | ： | ： |
| ただす 1 | ガ格 | 【主体: 氏, 委員, 議員】, 喚問 |
| | ヲ格* | 見解, 真意, 考え, 方針, 問題 |
| | について | 問題, <補文>, 展開, 責任 |
| ただす 2 | ガ格 | 【主体: 委員長, 自ら, 業界】 |
| | ヲ格* | 【主体: 身】, 姿勢, 姿, 威儀 |
| ： | ： | ： |
| 人気 1 | ガ格 | グッズ, 自転車, ベルト |
| | ニ格* | 【主体: 女性, 主婦, 男性】 |
| 人気 2 | ガ格* | もの, サービス, 商品, 海産物 |
| 人気 3 | ガ格* | <補文> |

52,000 個の用言について格フレームが構築され、用言あたりの平均格フレーム数は 1.3 個、格フレームあたりの格の平均数は 1.9 個、格あたりの平均異なり格用例数は 6.0 個であった。また、クラスタリングによって用例格フレーム数は用例パターン数の 38% になった。

構築した格フレーム辞書をみると、「人気」、「賛成」といった名詞+判定詞の格フレームも学習できている。また、「ただす」の「について」のように、複合辞の格についても学習できている。

## 7 解析実験

得られた格フレーム辞書の静的な評価は難しいので、それを用いた格解析を通して評価する。毎日新聞の記事 100 文をテストセットとし[2]、これに対して格解析を行った。格解析は [2] の方法を用いた。格解析結果の評価は、提題と被連体修飾詞の格を正しく認識できるかどうかで行う。格解析の評価を表 2、解析結果の例を次に示す[3]。

**(1)** 1大蔵省は○ガ格 九日、信託銀行の不良債権の処理を促進するため、一九九五年三月期決算で信託銀行各行が 2[積み立てている] 2特別留保金の○ヲ格 取り崩しを 3[認める] 3方針を×ニ格⇒ 外の関係 1[決めた。]

**(2)** 金権選挙追放策の一つとして、戦後 1[廃止されてしまった] 民衆訴訟による 1当選無効制度の○ガ格 2復活も ○ヲ格 2[試みる……。]

**(3)** これらの 1業界は×ヲ格⇒ ガガ、比較的外圧を受けにくく、また政治的発言力が強い、という特徴が 1[ある。]

**(4)** 代表質問を“ 影の内閣 ”として 1[設置した] 1政権準備委員会の×ニ格⇒ ガ格「施政方針演説」と位置付け、政権担当能力をアピールするのが狙い。

表 2 において、格解析の精度をみるために係り受けの誤りを除いて考えると、提題が 89%、被連体修飾詞が 73% というかなりよい精度で格の認識ができていることがわかる。誤りの大きな原因は、「〜を与える役割」のような外の関係、「業界は〜という特徴がある」といったガガ構文である。この問題の対処は今後の課題である。

## 8 おわりに

本論文では、用言とその直前の格要素の組を単位として、生コーパスから用例を収集し、それらのクラスタリングを行うことによって、格フレーム辞書を自動的に構築する手法を提案した。得られた辞書を用いて実際に格解析を行った結果、提題、連体修飾の格の解釈をかなり高い精度で行うことができた。従って、「使える」レベルの格フレーム辞書を構築できたと考えられる。今後、この格フレーム辞書を用いて文脈解析を行う予定である。

表 2: 提題、被連体修飾詞の格解析の評価

| | 正解 | 誤り | | | |
|---|---|---|---|---|---|
| | | 対応付けの誤り | 外の関係による誤り | ガガ構文による誤り | 係り受けの誤り |
| 提題 | 80 | 6 | – | 3 | 16 |
| 連体修飾 | 47 | 3 | 14 | – | 8 |

[2]このテストセットは、格フレーム辞書の構築には用いていない。

[3]○の下線部は格解析が正しく、×の下線部は誤っている。下線部の後に、格解析によって認識された格を記述し、格解析が誤っているときは、⇒ の後に正解の格を記述した。

## 参考文献


[1] Michael John Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of ACL*, pp. 184–191, 1996.

[2] S. Kurohashi and M. Nagao. A method of case structure analysis for japanese sentences based on examples in case frame dictionary. In *IEICE Transactions on Information and Systems*, Vol. E77-D No.2, 1994.

[3] NTT コミュニケーション科学研究所. 日本語語彙大系. 岩波書店, 1997.

[4] 宇津呂武仁, 宮田高志, 松本裕治. 最大エントロピー法による下位範疇化の確率モデル学習および統語的曖昧性解消による評価. 情報処理学会 自然言語処理研究会 97-NL-119, pp. 69–76, 1997.

[5] 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 自然言語処理, Vol. 1, No. 1, 1994.

[6] 情報処理振興事業協会技術センター. 計算機用日本語基本動詞辞書 IPAL (Basic Verbs) 説明書. 1987.

○ 研究開発プロジェクトの背景・経緯と目的

近年の社会活動全体のグローバル化（各国間の人的交流の活発化、ビジネス競争のボーダレス化、インターネットを介した外国語情報の大量流通、等）の進展に伴い、海外と外国語を通じて情報を受発信する機会が急速に増大している。このような状況で、日本語を外国語に翻訳したり、外国語を日本語に翻訳したりする機械翻訳技術や、様々な言語で書かれた文書の中から必要な情報を取得する情報検索技術の重要性がますます高まってきている。これら言語に関する技術は、近い将来の情報化社会建設の基盤となる重要な技術であり、我々の社会生活や企業活動に及ぼす影響は極めて大きい。

ところが現状では、機械翻訳、情報検索は実用化されているとは言え、いずれも十分といえるレベルに達していない。複数種類の言語をあつかうこれらの技術の水準を一層高める必要がある。たとえば機械翻訳は、現状では翻訳結果が直訳的であり、精度も５０％前後である。これを将来８０％を超える精度に高める必要がある。そのためのもっとも重要な技術は言語処理技術であるが、近年、優れた新技術が我が国の大学・研究機関で生まれている。例えば、自然言語文の構造を機械的に同定する技術、大量の文例（コーパスと呼ぶ）を利用する技術が開発されている。前者は東京工業大学、東京大学、京都大学で優れた手法が開発されている。後者は１９８０年代に京都大学・長尾教授によって提唱されたコーパスに基づく言語処理技術である。自然言語は、人為的に定められた人工物ではないため、その高精度な処理技術の開発には、大量の文例を収集し、それを詳細に分析・利用することが必須である。

コーパスから、翻訳に用いる対訳知識や、翻訳・検索に用いる語彙間の関係知識などを抽出することが可能になると、言語処理技術の性能が飛躍的に向上する。しかしながら、これまでは大量の電子化文書の入手が難しく、また前者の文構造を機械的に抽出する技術の水準も十分でなかったため、コーパスに基づく言語処理の研究の推進には困難があった。ところが近年、自然言語の電子化技術が急速に発展し、日本語・外国語の電子テキストが電子ネットワークを通して大量に流通するようになり、多言語大規模コーパスが利用可能となってくると共に、言語処理の基盤技術が実用レベルに到達してきた。

本共同研究では、大量のコーパスから、言語処理技術の高度化に役立つ知識を抽出し、インターネットの爆発的普及によりますます重要性を増している複数言語にまたがる言語処理応用システム（機械翻訳、情報検索等）の高度化に直接利用可能な知識を獲得する技術の研究開発を行うことを目的とする。

○ 共同研究組織

| | | |
|---|---|---|
| ・総括代表者 | 田中 穂積 | 東京工業大学 大学院情報理工学研究科 教授 |
| ・研究分担者 | 徳永 健伸 | 東京工業大学 大学院情報理工学研究科 助教授 |
| ・　〃 | 辻井 潤一 | 東京大学　大学院理学系研究科 教授 |
| ・　〃 | 黒橋 禎夫 | 京都大学 情報学研究科 講師 |
| ・企業分担代表者 | 亀井 真一郎 | 日本電気株式会社 情報通信メディア研究本部 主任研究員 |
| ・研究分担者 | 山端 潔 | 日本電気株式会社 情報通信メディア研究本部 主任研究員 |
| ・　〃 | 土井 伸一 | 日本電気株式会社 情報通信メディア研究本部 主任 |
| ・　〃 | 矢田部 清美 | 日本電気株式会社 情報通信メディア研究本部 研究員 |
| ・　〃 | 長田 誠也 | 日本電気株式会社 情報通信メディア研究本部 研究員 |
| ・企業分担代表者 | 梶　博行 | 株式会社日立製作所 中央研究所<br>マルチメディアシステム研究部 主任研究員 |
| ・研究分担者 | 森本　康嗣 | 株式会社日立製作所 中央研究所<br>マルチメディアシステム研究部 研究員 |
| ・研究分担者 | 小泉　敦子 | 株式会社日立製作所 中央研究所<br>マルチメディアシステム研究部 研究員 |
| ・企業分担代表者 | 松井 くにお | 富士通株式会社 ＤＢサービス部　担当部長 |
| ・研究分担者 | 潮田 明 | 富士通株式会社 ＤＢサービス部　担当課長 |
| ・　〃 | 橋本 三奈子 | 富士通株式会社 ＤＢサービス部　研究員 |
| ・　〃 | 富士 秀 | 富士通株式会社 ＤＢサービス部　研究員 |
| ・　〃 | 大倉 清司 | 富士通株式会社 ＤＢサービス部　研究員 |
| ・企業分担代表者 | 平川 秀樹 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　室長 |
| ・研究分担者 | 住田 一男 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　主任研究員 |
| ・　〃 | 木村 和広 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　研究主務 |
| ・　〃 | 大嶽 能久 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　研究主務 |
| ・　〃 | 木下　聡 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　研究主務 |
| ・　〃 | 小野 顕司 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　研究主務 |
| ・　〃 | 齋藤 佳美 | 株式会社東芝 研究開発センター<br>ヒューマンインターフェースラボラトリー　研究主務 |

合計　３５名

○ 研究期間 平成１２年3月１７日～平成１３年3月３１日

○ 研究開発の実施状況とまとめ

次頁より、各機関別の実施状況及びまとめについて記す。

電子・情報分野
９９Ｙ補０３－１１０－２

平成１１年度
新エネルギー・産業技術総合開発機構
提案公募事業（産学連携研究開発事業）
研究成果報告書

# 複数言語にまたがる言語知識処理技術の研究
# （対訳語彙知識抽出技術の研究）

平成１３年　３月

日本電気株式会社

**平成１１年度 新エネルギー・産業技術総合開発機構　提案公募研究開発事業**

**（産学連携研究開発事業） 研究成果報告書概要**

| | |
|---|---|
| 作成年月日 | 平成１３年3月３１日 |
| 分野／ﾌﾟﾛｼﾞｪｸﾄ ID 番号 | 分野：電子・情報分野<br>番号：９９Ｙ補０３－１１０－２ |
| 研究機関名 | 日本電気株式会社 |
| 研究代表者 部署・役職 | 情報通信メディア研究本部　　主任研究員 |
| 研究代表者名 | 亀井 真一郎 |
| プロジェクト名 | 「複数言語にまたがる言語知識処理技術の研究」<br>（対訳語彙知識抽出技術の研究） |
| 研究期間 | 平成１２年3月１５日 ～ 平成１３年3月３１日 |
| 研究の目的 | 大量の文例集（コーパス）から、言語処理技術の高度化に役立つ知識を抽出し、インターネットの爆発的普及により重要性を増している複数言語にまたがる言語処理応用システムの高度化に直接利用可能な知識を獲得する技術を研究開発する。特に本サブテーマでは、日本語と英語とで互いに翻訳関係にない同分野文例集（コンパラブル・コーパス）から、対訳知識を抽出する手法を開発することを目的とする。 |
| 成果の要旨 | コンパラブル・コーパスから対訳知識を抽出する手法の開発を行なった。日英それぞれのコーパスから、単語の出現確率、係り受け関係にある単語対の共起確率を求め、英日対訳辞書を用いて英語日本語間の単語の対応をとることで、対訳候補の対訳確率を推定することができ、この対訳確率が、そのコーパスが属する分野で確からしい訳語を選択するのに有効に作用することが実験により確認できた。 |
| キーワード | 機械翻訳、自動言語処理、自動辞書、自然言語、自然言語処理 |
| 成果発表・特許等の状況 | 特許出願１件： 特願２００１－１４４３３７ 「対訳確率付与装置、対訳確率付与方法並びにそのプログラム」 |
| 今後の予定 | 本プロジェクトで開発した対訳知識抽出手法およびその改良手法を組み込んで分野適応により翻訳精度を向上させた機械翻訳システムの製品化を２～３年後の目標とする。 |

Proposal-based R&D Program cooperated with Academic and Industrial organizations, in '99

| | |
|---|---|
| Date of preparation | March 31, 2001 |
| Field/<br>Project number | Field: Electronics and information technology<br>No. 99 Y HO 03-11-02 |
| Research organization | NEC Corporation |
| Post of the research coordinator | Principal Researcher<br>Computer and Communication Media Research |
| Name of the research coordinator | Shin-ichiro Kamei |
| Title of the project | `Multilingual Natural Language Processing Technologies'<br>(Research on extraction technology of translation equivalents from large scale corpora) |
| Duration of the project | March 15, 2000 ～ March 31, 2001 |
| Purpose of the project | Multilingual natural language processing is one of the most important technologies, since a large volume of multilingual texts are being created, circulated, and accumulated through Internet. This research develops a new method of extracting language information from large corpora, improving current natural language processing technologies. In particular, this subproject focuses to develop a method of extracting translation knowledge from comparable corpora. |
| Summary of the results | We developed a method to extract translation probabilities of a word in the source language into its translation equivalent in the target language. This method is based on syntactic and statistical analyses of comparable corpora, utilizing bilingual dictionaries. |
| Key Word | Natural Language Processing, Machine Translation, Comparable Corpus, Translation Equivalents |
| Publication, patents, etc. | Patent application No. 2001-144337 "Translation probabilities assigning device, method, and program" |
| Future plans | The developed method and its improvement will be implemented on a machine translation system in 2 or 3 years, of which quality is improved using domain dependent translation equivalents selection. |

[まえがき]

近年インターネット上のコンテンツの量が爆発的に増大し、多言語化が急速に進展している。ここから有用な情報を効率的に探し出し活用するためには、日本語や英語を処理する自然言語処理技術は必須の基幹技術である。

機械翻訳システムを例にとれば、現在のシステムは訳文の可読率がほぼ６割と言われている。精度向上のボトルネックは、システムがもつ辞書や規則など言語知識の質・量と、処理対象の広さとのギャップにある。適用分野を限定することでシステムの精度が大きく向上することはよく知られているが、このためには、分野別の言語知識を手で作り込む必要があり、多大なコストがかかるため、一般的には現実的な選択肢ではない。

この困難を解決するため、大規模な実文例（コーパス）からの言語知識の獲得技術の研究開発を行なった。特に、完全な対訳関係にはないが共通の内容について記述した日本語・英語の文例集（コンパラブル・コーパス）から、日本語・英語の対訳知識を抽出する手法の開発を行なった。開発した手法は、日英それぞれのコーパスの中から、共起関係、係り受け関係にある単語対を大量に抽出し、日英対訳辞書を用いて日英の単語対間の対応をとることで、対訳辞書の対訳候補の中から、そのコーパスが属する分野で確からしい訳語を選択するものである。

[研究者名簿]


研究代表者　亀井 真一郎　日本電気株式会社 情報通信メディア研究本部
音声言語テクノロジーグループ　主任研究員

研　究　者　山端　潔　日本電気株式会社 情報通信メディア研究本部
音声言語テクノロジーグループ　主任研究員
土井 伸一　日本電気株式会社 情報通信メディア研究本部
音声言語テクノロジーグループ　主任
矢田部 清美　日本電気株式会社 情報通信メディア研究本部
音声言語テクノロジーグループ　研究員
長田 誠也　日本電気株式会社 情報通信メディア研究本部
音声言語テクノロジーグループ　研究員

[目次]

## １：研究の背景

機械翻訳を始めとする異なる言語を結ぶ自然言語処理を高度化する手がかりとして、近年「コンパラブル・コーパス」が注目されてきている。「コンパラブル・コーパス」とは、互いに対訳関係にはないが、同じ事柄について日本語と英語など異なる幾つかの言語で書かれた文章の集まり(文例集)のことを言う。本プロジェクトでは、機械翻訳などにおける訳語選択を行なうための言語知識を、コンパラブル・コーパスから半自動的に抽出する基盤技術の研究開発を行なった。

近年インターネット上のコンテンツの量が爆発的に増大し、多言語化が急速に発展している。ここから有用な情報を効率的に探しだし活用するためには、日本語や英語を処理する自然言語技術は必須の基幹技術である。機械翻訳システムを例にとれば、現在のシステムは訳文の可読率がほぼ６割と言われている。精度向上のボトルネックは、システムがもつ辞書や規則など言語知識の質・量と、処理対象の広さとのギャップにある。適応分野を限定することでシステムの精度が大きく向上することはよく知られているが、このためには、分野別の言語知識を手で作り込む必要があり、多大なコストがかかるため、一般的には現実的な選択肢ではない。

従来の機械翻訳システムは、翻訳対象の文書を、それ単独として処理してきた。しかし、あらかじめ、多種類のジャンルの文書を用意してその傾向を分析しておき、翻訳対象の文書がどのジャンルに近いかを判定することができれば、用いられている単語の認定もその訳語の選択も、今より格段に適切なものになることが期待される。このような、コーパス(文例集)を使った翻訳品質の向上が期待できるようになったのは、今までの長い時間と多大な努力をかけた機械翻訳の研究開発の結果、基礎的な文解析の精度や速度が非常に高くなったこと、基盤となる基本対訳辞書が整備されたことなど、技術の発達・蓄積があったからである。これに合わせて、大量の電子テキストが流通・蓄積され、利用可能になったため、分析技術、分析対象の両方がそろい、まさに現在、大規模コーパスから言葉の知識を獲得する研究の環境が整いつつある。そこで本プロジェクトでは実文例(コーパス)からの言語知識獲得の基盤となる技術の研究開発を行なった。

現在の機械翻訳は、購入してすぐに十分に使えるシステムではない。使用者は、自分が翻訳したい文書の中に、機械翻訳システムに登録されていない単語があれば、自分でその単語とその単語の訳語とを機械翻訳の辞書に登録しなければならない。またシステムの出力する訳語が適切でなかった時には、訳語を選び直したり適切な訳語を追加したりする必要がある。このように、ユーザが自分でシステムに手を加えていかなければ望ましい結果が得られない、という点は、購入してスイッチを入れればすぐに使える従来の電気製品と機械翻訳のような言語処理システムとの最も大きな違いといえる。

例えば「stock」という英単語は、経済用語として「株」を意味するが、辞書を引くと、「貯蔵」「蓄え」「うんちく」「（農業分野で）家畜」など、他にもたくさんの意味をもって

いる。現在の多くの翻訳システムにとっては、このような、意味のたくさんある単語の訳語を適切に選ぶことは非常に難しい。株価の文章を翻訳しようと思っているのに「貯蔵」と訳出されてしまったり、経済の文章ではないのに「株」と出てきてしまったりといった、不適切な結果になってしまいがちなのが現状である。

機械翻訳は、もちろん他にも沢山の困難な側面をもっているが、訳語選択の問題は中でも最も重要な大問題といえる。翻訳対象の文章のジャンルが経済なら経済と狭い範囲に決まっていれば、ジャンルごとに用いられる用語が定まるので、このような不適切な訳出の問題もかなりの程度減らすことがが、一般に使用者が対象とするジャンルをあらかじめ定めることはできない。そこで現状の多くのシステムは、いかなる場合にもかなりの程度に妥当な訳を与えるというところまでには至っていないのが現状である。したがって現状では、システムの使用者が、単語の訳語を選び直して適切なものにする必要がある。

しかしながら、システムの使用者が自分で訳語を選ばなければならないという制約は、一般の使用者には大きな負担である。使用者は、訳し方がわからないからこそ機械翻訳を使いたいのであろう。にもかかわらず、「適切な翻訳結果が欲しければ、自分で訳語を選択せよ」という負担を強いられるのは、使用者にとって矛盾に感じられる。この負担を減らすことができれば、機械翻訳システムは現状よりも格段に普及すると思われる。そこで現在、このような使用者の負担を軽減させるべく、訳語選択の精度向上の研究がすすめられている。

## ２：コーパスからの言語知識獲得に関する従来研究

本プロジェクトでは、訳語選択の精度向上へのアプローチとして、コーパスを用いた訳語選択の手法、具体的には、コンパラブル・コーパスからの訳語選択知識抽出手法の研究開発を行なった。本節では、本研究に先立つ、コーパスからの言語知識獲得に関する従来手法について述べる。

機械翻訳、クロス言語テキスト検索など、異なる言語の間で言葉の対応をとることを課題とする自然言語処理技術においては、一方の言語の単語を、もう一方の言語の適切な単語に対応させることは非常に重要な課題であり、訳語選択の問題と呼ばれている。例えば、英語の単語は一般に複数の意味をもち、一般にはそれぞれ異なる日本語の単語に対応する。ところが、自然言語処理分野において元の英単語の使われている状況を正しく判断して適切な日本語の単語を選択することは、一般には非常に困難である。例として英語の単語「term」には「期間」という意味の他に「専門用語」という意味があるが、どのような場合に「期間」という意味となり、どのような場合に「専門用語」という意味になるか、という訳語の選択条件を、あらかじめ明示的に記述することは非常に難しい。

使われる単語やその単語に対する訳語は、政治、経済、科学、料理、スポーツ、芸術、といった分野ごとに大きく異なる。そういった分野は非常に数多く存在するから、日本語や英語の単語や訳語を、あらかじめすべてシステム上に用意しておくことは困難である。しかし、だからといって、そういった単語の登録や訳語の指定をシステム使用者自身が行わなければならない、という現状の負担を軽減してゆかなければ、機械翻訳が真に活用されるようにはならず、グローバルネットワーク社会で大量に流通する多言語情報を活用した生活を営むことは不可能である。そこでそのようなユーザの負担を軽減する方法として、いくつかの試みが始まっている。

そのひとつの方法として着目されているのが、類似した文章を大量に集めそこから言葉の知識を獲得する技術である。システム使用者が翻訳したい文章と類似したジャンルの文章が大量にある状況を想定する。そのような状況では、その文章を分析することで、そのジャンルで頻繁に使われる単語や言い回しを抽出することができる可能性がある。またそのジャンル特有の訳出法の傾向も半自動で抽出できることが期待される。このような知識をシステムが翻訳に有効に利用できれば、現在より翻訳の品質が上がり、システム使用者の負担を軽減できる可能性がある。このような期待から、大量の文章、すなわち「コーパス」の研究が盛んに行われている。

### （１） 目的言語だけを使った訳語選択

コーパスを使った言語知識獲得の技術は、人の手による言語知識の抽出の限界が認識され始めた１９９０年代初めから研究が開始された。しかし研究の初期には、利用できるコー

パスがまだ少量であったこと、コーパスの言語的加工に必要となる言語処理の基盤技術がまだ充分成熟していなかったことなどから、コーパス研究は様々な制限の中で行なわれ始めた。

たとえば 文献[1] [2]では、次のような、主として目的言語コーパスだけを用いる方法が提案されている。まず、ある言語(日本語など)から第２の言語(英語など)への翻訳を行なう場合、同じ分野の話題を述べている第２の言語の文例を大量に収集しておく。元の言語の単語が、相手言語の訳語候補のうち、どの訳語に対応するかの確からしさを判定する際に、相手言語の文例集における、各訳語候補の出現確率の高さを用いる。たとえば、今、英語の「term」を「期間」と訳すのが確からしいか「専門用語」と訳すのが確からしいかを判断するのに、同じ分野の話題を述べている日本語の文例集の中に出現する「期間」という単語と「専門用語」という単語の頻度を計測し、その多い方を「term」の訳語とする、という手法である。この手法には、相手言語の文例集のみを分析すればよいという利点がある。

しかしながら、この方法は、相手言語の単語の出現傾向だけを手がかりにしているため、相手言語で一般的に高頻度で出現する単語が訳語として採用されてしまいやすい、という欠点がある。たとえば、英語の単語「fruit」には「果物」という訳語の他にも多くの日本語の訳語が相当する。一例として物事の成果を果物にたとえる場面では、「fruit」には「結果」「成果」といった訳語が対応する。この場合、従来の方法に従って、相手言語、つまり日本語の単語の出現頻度だけを計測すると、fruit に対する最も一般的な訳語「果物」よりも訳語「結果」の方が一般に使用頻度が高いので、「fruit」の訳語候補として「結果」が最も確からしいものとして選択されてしまう危険がある。従来の目的言語のみを用いて訳語選択を行なう方法には、このように、本来の訳語として適切かどうかとは無関係に、相手言語で出現頻度の高い訳語が選択されやすい、という欠点があった。

## （２）パラレルコーパスを使った訳語選択

翻訳の品質を上げるには、日本語の語彙・表現と英語の語彙・表現とをスムーズに対応させるためのデータや傾向や規則を大量に必要とする。そのようなデータや傾向や規則を得るには、日本語と英語が対応しているコーパスが必要である。同一の内容について、日本語と英語のような異なる言葉で、文のレベルまでほぼ対応して書かれているコーパスを「パラレル・コーパス」と呼ぶ。例えばカナダでは英語話者とフランス語話者の両方に同一の情報を提供する必要性から、同一の内容を、英語とフランス語の２カ国語で書いたパラレル・コーパスが存在する。このようなパラレル・コーパスが大量にあれば、ある単語がどのように訳されるか、といった傾向を分析して機械翻訳に応用することが可能となる。そのためパラレル・コーパスの研究は非常に重要である。日本語の場合でも、例えば、外国に輸出する製品の操作マニュアルなどは、日本語と外国語の両方がかなり厳密に対応し

た文章で書かれているが、これは非常によく対応したパラレル・コーパスといえる。このような対応のよくとれたパラレル・コーパスを対象として、翻訳精度向上のための語彙・表現や訳語のデータを抽出し、それを利用する研究が続けられている(文献[4])。代表的な手法は以下のような方法である。まず、異なる言語(日本語と英語など)で、一方が他方の翻訳関係にあるような対訳文例集（コンパラブル・コーパス）を用意する。さらに、二つの言語間の対訳辞書を用意し、第１言語の例文に含まれる単語に対して対訳辞書を引き、訳語候補を挙げる。その例文と対訳関係にある第２言語の例文の中に出現する訳語候補の頻度を計測し、最も高頻度で現われる訳語候補を、元の単語に対する訳語とする。この手法は、互いに翻訳関係にある対訳例文が利用できる場合には、高い精度で訳語を認定できるという利点がある。

このような方法は、互いに翻訳関係にある対訳例文が大量に存在する場合に有効な方法であるが、実際には、互いに翻訳関係にある対訳例文(パラレル・コーパス)の量は極めて限られている。パラレル・コーパスを前提とする方法には対訳例文が大量に存在しない場合には適用できない、という問題点がある。

## （３）コンパラブル・コーパスへの期待の高まり

上述したようにパラレル・コーパスからの翻訳知識の獲得は非常に有益である。しかしながら、残念なことに、日本語と他の言語の間のパラレル・コーパスは、一般にはあまり多くは存在しないのが現実である。パラレル・コーパスは大量に存在しないと語彙や訳語を有効に抽出することができない。上述した製品マニュアルの他には、公的な文書の一部や、新聞記事の一部などにパラレル・コーパスが存在するが、個々のユーザが翻訳したいと思う多種・多様なジャンルの文章には、ほとんどの場合、大規模なパラレル・コーパスは存在しないのが現状である。

そこでパラレル・コーパスに代わるものとして、「コンパラブル・コーパス」が注目されはじめた。ここで「コンパラブル」という言葉は「比較しうる」とか「対応しうる」といった意味で使われている。つまり「コンパラブル・コーパス」とは、互いに翻訳関係にはなく、その意味で厳密な対応はしていないが、同種の内容を異なる言葉で書いた文章の集まりのことを指す。例えば、ある一つの出来事について日本語と英語とで書かれた新聞記事は、それが翻訳関係にある場合には、パラレル・コーパスだが、たとえ翻訳関係でなく独立に書かれた場合であっても、同一の出来事について書いてある記事である以上、かなりの程度、語彙や表現に対応があるものと思わる。これがコンパラブル・コーパスの典型例である。

このようなコンパラブル・コーパスが大量にあれば、単語や言い回しの対応の確からしさも高まり、そこから翻訳に有効な知識が抽出できることが期待される。例えば、株式に関する日本語と英語の文章が大量にあれば、英語の文章には「stock」という単語が頻繁に

使われているだろうことが想像される。一般には「stock」に対応する日本語の単語はたくさんあり得るが、このジャンルの日本語の文章の中では、その中で「株」の頻度が高いことがわかる。「株」と「stock」が対応していることを確かめるには、使われている頻度だけではなく、その文章の文脈をあらわす手がかり、例えば同じ文中に一緒に使われている単語や、修飾関係にある単語など、いろいろな条件を考慮して判断する必要があるが、そのような判断も、コンパラブル・コーパスが大量にあれば、信頼性が高まる。

厳密な翻訳関係による文の対応づけがなくてもよいなら、同様の事柄について異なる言語で書かれた文章は大量に存在する。グローバルネットワーク上に電子テキストがあふれている現在では、ほとんどのジャンル、例えば、料理であれ、スポーツであれ、音楽であれ、書かれている内容がかなりの程度重なる文章を見いだせる可能性がある。それを見いだし、そこから翻訳に有効な知識を得ることができるのではないか。コンパラブル・コーパスが注目されている理由である。

コンパラブル・コーパスの可能性は９０年代半ばごろから研究されはじめた(文献[3][5][6][7][8])。しかし当時は、入手できるコンパラブル・コーパスの量が限られていたので本格的な実証研究にまでは至っていなかった。最近になって、大量の多種類・多言語の電子テキストが利用可能になってきたため、コンパラブル・コーパスからの知識抽出の現実性が急速に高まってきた(文献[9][10][11])。本研究は、このような流れの中で、機械翻訳やクロス言語情報検索にとって特に重要な訳語選択知識をコンパラブル・コーパスから抽出する手法を研究開発することが目的である。

## ３：提案手法：コンパラブル・コーパスを用いた訳語選択

我々は、元言語および目的言語に出現する各単語についてのパラメータを含む対訳確率モデルを定式化した。この対訳確率モデルと実際のコーパスから求められる統計量との差分を最小にするようにパラメータを定める。対訳確率モデルは次のように表現できる。

＜モデル＞
元言語の各単語の出現確率と、元言語の各単語と意味的に対応する
目的言語の各単語の出現確率とは、一致する。

このモデルでいう出現確率を求めるための制約としては、それぞれの単語を独立に考えた場合の出現制約だけでなく、係り受け関係などにある単語対の出現制約、さらに高次の共起関係にある単語セットの出現制約からなるものと考える。これを模式的な式で表すと次のようになる。

単語の出現確率を求めるための制約

$$= \sum_{N=1} \text{N次の共起関係を考慮した単語の出現制約}$$

＝　単語単独の出現制約
＋　共起関係にある２単語の出現制約
＋　共起関係にある３単語の出現制約
＋　共起関係にある４単語の出現制約
＋　・・・・・　　式（１）

ここで１次の共起関係とは単語単独の出現確率、２次の共起関係とは、一文内共起、隣接、あるいは係り受け関係などの関係にある２単語の出現確率、等々を表すものとする。実際の計算では、N の何次までを用いて近似するかが選択できる。

提案モデルは、元言語および相手言語においてこの出現確率が対応することを仮定する。これを模式的に表すと次式のようになる。

元言語における単語の出現確率　＝　相手言語における単語の出現確率
式（２）

次に、元言語の各単語と相手言語の単語が意味的に対応する、という内容を具体的に定式化するために、対訳確率モデルの各展開項の具体形を考える。次式は、対訳確率モデルの第１近似、すなわち、単語単独の出現確率（対訳確率モデル式（１）の第１項）が、元

言語と相手言語とで対応することを表す式である。

$$j(m) = \sum_{i} e(i) \times S(i,m) \qquad \text{式(3)}$$

この式において、e(i) は第１言語(例えば英語)の i 番目の単語 E(i)の出現確率を表す。また j(m) は第２言語(例えば日本語)の m 番目の単語 J(m) の出現確率を表す。S(i,m) は、第１言語の i 番目の単語 E(i)が、第２言語の m 番目の単語 J(m)に翻訳される確率を表す。この式は、第１言語の各単語の出現確率と翻訳確率の積の総和が第２言語の各単語の出現確率を与えるというモデルを表している。この式において、翻訳確率 S(i,m) が、この対訳確率モデルにおけるパラメータであり、第１言語の単語 E(i) と第２言語の訳語候補 J(m) との単語対応対に与えられた対訳確率である。このパラメータには、第１言語の単語は第２言語の単語に必ず対応するという仮定の下で、

$$\sum_{m} S(i,m) = 1 \qquad \text{式(4)}$$

という制約がある。本来は、第１言語の１単語が第２言語の複数の単語に対応したり、第１言語の複数の単語が第２言語の１語に対応したり、という対応単語数のずれがありうる。しかし本モデルでは、単語対単語の対応がよい近似で成立するものという仮定をおいた。

図１は、対訳辞書における元言語と相手言語の訳語関係を表わしている。この図では、第１言語の単語 E(i) に対応する第２言語の訳語候補として、J(m)、J(n)、J(p) が存在する場合を示している。この図で e(i)は、第１言語の単語 E(i) の出現確率、j(m)、j(n)、j(p) はそれぞれ第２言語の単語 J(m)、J(n)、J(P) の出現確率を表す。また、S(i,m)、S(i,n)、S(i,p) はそれぞれ、第１言語の単語 E(i) が、第２言語の単語 J(m)、J(n)、J(p) に翻訳される確率を表す。

この対訳確率モデルによって各単語の対訳確率を求めるには以下の計算を行なう。次式は式（３）を左辺に移項して自乗したものである。

$$\left( j(m) - \sum_{i} e(i) \times S(i,m) \right)^2 \qquad \text{式(5)}$$

本手法では、第１言語の文例集における統計量を計算して第１言語の単語の出現確率 e(i) を計算し、同様に、第２言語の文例集の統計量を計算して第２言語の単語の出現確率 j(m) を計算する。このようにして求めた e(i)および j(m)を式（５）に代入し、上記式（４）の制約を満たす条件の下で式（５）の値を最小にするようなパラメータ S(i,m) を定める。こ

のようなモデルのパラメータ・フィッティングによって定めたパラメータ S(i,m) の値が対訳確率を与えると考える。

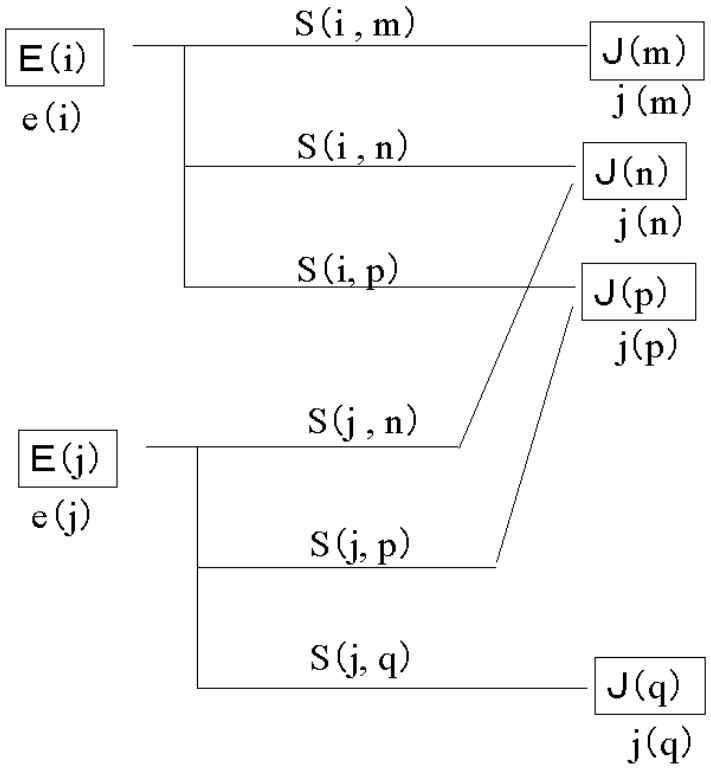



**図１　対訳辞書の見出しの出現確率と対訳確率**

次式は、共起関係にある２単語の出現制約（対訳確率モデル式（１）の第２項）が元言語と相手言語とで対応すること表す式である。

$$P(J(m)\wedge J(n)) = \sum_{i,j} P(E(i)\wedge E(j)) \times S(i,m;j,n) \qquad \text{式(6)}$$

この式において、P(E(i)^E(j)) は、第１言語で単語 E(i) と単語 E(j)が同時に出現する共起確率を表し、P(J(m)^J(n)) は、第２言語で単語 J(m) と単語 J(n) が同時に出現する共起確率を表す。また S(i,m；j,n) は、単語 E(i)が単語 J(m)に対応しかつ単語 E(j)が単語 J(n)に対応する共起対訳確率を表す。この式は、第１言語における二つの単語の共起確率と対訳確率の積の総和が、第２言語における二つの単語の共起確率を与えるというモデルを表し

ている。この第２項までを考慮したモデル・パラメータの計算式は以下のようになる。

$$\alpha \left( j(m) - \sum_{i} e(i) \times S(i,m) \right)^2 + \beta \left( P(J(m)\wedge J(n)) - \sum_{i,j} P(E(i)\wedge E(j)) \times S(i,m;j,n) \right)^2$$

式（７）

式（７）は、式（６）を左辺に移項して２乗したものと、式（５）の線形和である。本手法では、コーパスから得られた統計量を代入してこの式の値を最小にするように対訳確率パラメータを定める。式（５）と同様に、第１言語の文例集における統計量を計算して第１言語の単語の出現確率 e(i) および単語の共起確率 P(E(i)^E(j)) を計算し、同様に、第２言語の文例集の統計量を計算して第２言語の単語の出現確率 j(m) および単語の共起確率 P(J(m)^J(n)) を計算する。このようにして求めた e(i)、j(m)、P(E(i)^E(j))、P(J(m)^J(n))を式（７）に代入し、この式の値を最小にするようなパラメータ S(i,m) および S(i,m；j,n) を定める。ここでα、βは、対訳確率モデルの式の第１項と第２項の寄与の割合を表すパラメータであるが、このパラメータの設定の方法については後述する。

以上、本手法の対訳確率モデル式（１）の第２近似までの計算方法を述べた。第２近似とはすなわち、単語の出現確率を求めるための制約として、個々の単語単独の出現制約と、共起関係にある２単語の出現制約までを考慮した近似である。本手法の対訳確率モデル式（１）は３単語以上の出現制約も取り込んで近似を上げることができる方法となっている。

## ４：実験

本節では、前節で提案した対訳確率モデルに基づく訳語選択方式の実証実験について述べる。本方式は、どのような言語対においても適用可能であるが、今回は、元言語として英語を、相手言語として日本語を選び、英日対訳の訳語選択について実験を行なった。図２は本提案方式の実験システムの構成を示す。第１言語文例集は、第１の言語、すなわち英語の実例文集（英語コーパス）である。第２言語文例集は、第２の言語、すなわち日本語の実例文集（日本語コーパス）である。対訳辞書には、英語の各単語に対する日本語の訳語候補を単語対応対として格納してある。

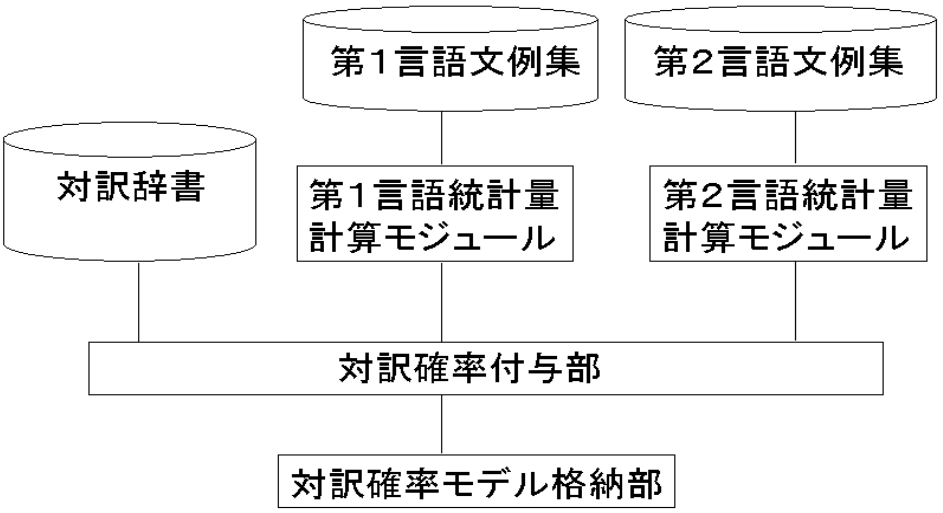



図２　　実験システムの構成図

第１言語統計量計算モジュールは、第１言語すなわち英語コーパスにおける単語の出現に関する統計量を計算する。第２言語統計量計算モジュールは、第２言語すなわち日本語コーパスにおける単語の出現に関する統計量を計算する。第１、第２言語統計量計算モジュールは、必要に応じて、英語コーパス、日本語コーパスに含まれる文を形態素解析したり構文解析したりして、そこに含まれる単語の出現に関する統計量を計算する。

対訳確率モデル格納部には、前節で述べた対訳確率モデルが格納してある。この対訳確率モデルは、対訳辞書の各単語対応対に付与された対訳確率をパラメータとして、英語コ

ーパスから求められる統計量から、日本語コーパスの単語の出現に関する統計量を推定する。

対訳確率付与部は、日本語コーパス求められた統計量と、上記の対訳確率モデルによって英語コーパスを使って推定された日本語の統計量との差を最小にするように、対訳辞書の各単語対応対に付与された対訳確率パラメータを調整する。

具体的には、対訳確率の計算式としては、第１近似として式（５）を用いた。

$$\left( j(m) - \sum_{i} e(i) \times S(i,m) \right)^2 \qquad \text{式(５)}$$

また、第２近似としては、式（７）を元にさらに近似をほどこした式（８）を用いた。

$$\left( j(m) - \sum_{i} e(i) \times S(i,m) \right)^2 + \left( P(J(m)\wedge J(n)) - \sum_{i,j} P(E(i)\wedge E(j)) \times S(i,m) \times S(j,n) \right)^2$$

式（８）

この式（８）は、式（７）において、共起対訳確率 S(i,m；j,n) を単語対訳確率 S(i,m)および S(j,n) の積で近似したものである。かつ、第１項と第２項の寄与（α、β）は１：１に設定した。

この対訳確率モデルの第２近似（式（８））を使って各単語の対訳確率を求める場合、採用する単語共起としては何種類かが考えられるが、本実験では、二つの単語が互いに構文的な係り受け関係にある係り受け共起の場合の効果について実験を行なった。

上記式（５)、式（８）における S(i,m)および S(j,n)が、この対訳確率モデルにおけるパラメータであり、それぞれ、英単語 E(i) とその日本語訳語候補 J(m) との単語対応対に与えられた対訳確率、英単語 E(j) とその日本語訳語候補 J(n) との単語対応対に与えられた対訳確率である。出現した英単語は日本語の単語に必ず対応するという仮定の下で、このパラメータには、

$$\sum_{m} S(i,m) = 1 \qquad \text{式(４)}$$

という制約がある。

英語コーパスを解析して、各単語の出現確率 e(i)　および二つの単語の係り受け共起確率 P(E(i)^E(j)) を求め、日本語コーパスを解析して、各単語の出現確率 j(m)　および二つの単語の係り受け共起確率 P(J(m)^J(n)) を求める。このようにして得られた e(i) 、j(m)、P(E(i)^E(j)) 、P(J(m)^J(n)) を上記の対訳確率モデルの式に代入して、上記の制約式（４）を満たすパラメータ S(i,m) を定める。また最急降下法を用いて式（５）および式（８）の最小値を求めた。

実験にあたっては以下の言語資源を用いた。日本語テキストの解析に際しては、京都大学情報学研究科黒橋研究室で開発された日本語形態素解析システムＪUMAN(ver.3.61)、構文解析システムＫＮＰ(ver.2.0b6)、および東京工業大学大学院情報理工学研究科田中研究室・徳永研究室で開発された日本語解析システムＭＳＬＲ Parser(ver.1.03)を使用した。英語テキストの解析に際しては、東京大学大学院理学系研究科辻井研究室で開発されたＸＨＰＳＧパーサー、およびニューヨーク大学コンピュータ科学科で開発された Apple Pie Parser (ver.5.9)を使用した。各言語の解析システムを２種類ずつ用いたのは、係り受け単語対を求める際に、個々の解析システムの解析誤りの傾向が結果に強く反映するのを避けるためである。

英日対訳辞書としては、日本電子化辞書株式会社で開発されたＥＤＲ電子化辞書 英日対訳辞書 ver1.5 を利用した。英語および日本語の大規模コンパラブルコーパスとしては、以下の２種類、すなわち読売新聞社から提供を受けた新聞データと、当社独自に収集した旅行会話に関するコーパスとを用いた。

(１) 読売新聞コーパス (日本語、英語 各１年分(１９９９年)；
日本語約２１万文、英語約１２万文)
(２) 旅行会話コーパス (独自収集；日本語、英語各約２万文)

実験では、英日翻訳における訳語選択を想定して英日対訳辞書を用いて実験を行なった。英語、日本語それぞれのコーパスで単語の出現に関する統計データを求め、前述の式（５）および式（８）に基づいて、パラメータすなわち対訳確率 S(i, m) を求めた。

## ５：結果と考察

### ５－１：コーパスの解析結果

実験に使った２種類のコーパス、すなわち新聞記事１年分と旅行会話文を解析した結果の単語数および係り受け関係にある単語対数は以下の通りである。実験では、コーパスに出現した単語の中で EDR 辞書に存在する単語を対象にした。また新聞記事データに関しては、出現頻度が６回以上の単語のみを分析対象とした。

[新聞記事コーパス]

| | 解析結果が得られた文数 | 全出現単語数 | 頻度 6 以上で EDR 辞書に存在する単語数 | 係り受け単語対の異なり数 | 頻度 6 以上でＥＤＲ辞書に単語が存在する係り受け単語対数 |
|---|---|---|---|---|---|
| 英語 | 12 万文 | 13,438 | 3,668 | 169,383 | 100,178 |
| 日本語 | 21 万文 | 89,406 | 7,288 | 1,109,297 | 312,844 |

[旅行会話コーパス]

| | 解析結果が得られた文数 | 全出現単語数 | EDR 辞書に存在する単語数 | 係り受け単語対の異なり数 | EDR 辞書に単語が存在する係り受け単語対数 |
|---|---|---|---|---|---|
| 英語 | ２万文 | 3,587 | 2.101 | 22,916 | 14,245 |
| 日本語 | ２万文 | 6,554 | 2,426 | 28,122 | 10,344 |

### ５－２：対訳モデルの実験結果

表１に今回の対訳モデルを使った実験結果の一部を示す。表１において、１行目（モデル欄「新聞０」）は新聞記事１年分を対象にして計測した目的言語（日本語）の出現頻度上位の訳語候補を示している。２行目（モデル欄「旅行２」）は、旅行会話コーパスを対象にして、提案手法の式（８）（モデル第２近似）を計算して求めた上位の訳語候補を示している。

この表から、一見して、２行目で旅行に特徴的な訳語が有効に選択されていることがわかるが、詳細にみると、いくつかのグループがある。まず、１行目では目的言語すなわち日本語における頻度情報のみから訳語候補を求めていることによって、不適切な訳語が第１候補に挙げられていたのが、本手法の第２行目で妥当な訳が選ばれるようになっているグループである。この例としては、arm(力 → 腕)、 fruit(結果 → 果物)、 ticket(メモ → 切符)　などが挙げられる。これらは、１行目では、目的言語の頻度のみを用いたために、

訳語としてはあり得るが実際には派生義であってその単語の第１訳語としては不適切な語が選ばれてしまっている例である。これらの例で本手法によって第１候補になった語は、旅行会話という分野に依存しているというより、より広く一般的な語といえる。これらの例からも、目的言語のみを用いる手法より、元言語と目的言語の両方のコーパスを用いる手法の法が妥当な結果を出すことがわかる。

次に特徴的なグループは、１行目の第１候補としてはその英単語の訳語としては最も妥当だと思われるが、２行目では全く別の語が候補として選ばれているグループである。その例としては、gas(ガス → ガソリン)、number(数 → 番号)、straw(わら → ストロー)等が挙げられる。これらの例の場合、新聞記事１年分の中に出現した最大頻度の候補は、その英単語の候補として一般には何ら問題がない。しかし、現実の場面では、それとは異なった訳語が最も妥当であるような例である。

また、新聞が書き言葉なのに対して、旅行会話が口語的であることによる差異を生じているグループがある。たとえば、beverage(飲料 → 飲み物)、 food(食品 → 食べ物)、today(現在 → 今日)等は、意味的には同等か非常に近い訳語同士であるが、用いたコーパスの文体の差が訳語候補に反映している。

残りの例の多くは、一般的な訳語から旅行会話特有の訳語に変化したグループである。たとえば、duty(義務 → 関税)、extension (拡大 → 内線)、prescription(規定 → 処方箋)等は、具体的な旅行場面で実際に使われる訳語が選ばれるようになっている。

これらの結果から、本提案モデルは、目的言語のみを用いる手法の欠点を克服しつつ、分野適応した訳語を優先させることのできる手法であることがわかった。

[表１] 新聞記事：目的言語のみの計算方法と、旅行会話：提案モデルの計算方法の比較

| 英単語 | モデル | 日訳語 １位候補 | 翻訳確率 | 日訳語 ２位候補 | 翻訳確率 | 日訳語 ３位候補 | 翻訳確率 |
|---|---|---|---|---|---|---|---|
| absolutely | 新聞０ | 全く | 0.840426 | 全然 | 0.159574 | | |
| | 旅行２ | 全然 | 0.778906 | 全く | 0.221094 | | |
| address | 新聞０ | 講演 | 0.319295 | 演説 | 0.315377 | 住所 | 0.264447 |
| | 旅行２ | 住所 | 0.811620 | 番地 | 0.125911 | アドレス | 0.030158 |
| age | 新聞０ | 時代 | 0.541295 | 期 | 0.189065 | 世代 | 0.095540 |
| | 旅行２ | 年齢 | 0.285761 | 長い間 | 0.284947 | 期 | 0.143531 |
| application | 新聞０ | 適用 | 0.291840 | 申し込み | 0.230982 | 応用 | 0.141079 |
| | 旅行２ | 申し込み | 0.666676 | 適用 | 0.333324 | | |
| arm | 新聞０ | 力 | 0.652394 | 備える | 0.121547 | 武器 | 0.066759 |
| | 旅行２ | 腕 | 0.714640 | 袖 | 0.285360 | | |
| beverage | 新聞０ | 飲料 | 0.808511 | 飲み物 | 0.191489 | | |
| | 旅行２ | 飲み物 | 0.956826 | 飲料 | 0.043174 | | |
| conductor | 新聞０ | 指揮者 | 0.882353 | 車掌 | 0.117647 | | |
| | 旅行２ | 車掌 | 1.000000 | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| direction | 新聞０ | 方針 | 0.350959 | 管理 | 0.233973 | 方向 | 0.095616 |
| | 旅行２ | 案内 | 0.803701 | 方向 | 0.117548 | 指示 | 0.034936 |
| duty | 新聞０ | 義務 | 0.239875 | 税 | 0.222741 | 効率 | 0.140187 |
| | 旅行２ | 関税 | 0.668363 | 義務 | 0.331637 | | |
| exchange | 新聞０ | 交換 | 0.354603 | 為替 | 0.258368 | 取引所 | 0.132845 |
| | 旅行２ | 両替 | 0.415504 | 交換 | 0.228715 | 小切手 | 0.153357 |
| extension | 新聞０ | 拡大 | 0.572115 | 範囲 | 0.245192 | 広がり | 0.068269 |
| | 旅行２ | 内線 | 0.909482 | 範囲 | 0.090518 | | |
| food | 新聞０ | 食品 | 0.354839 | 食糧 | 0.224014 | 食べ物 | 0.136201 |
| | 旅行２ | 食べ物 | 0.746096 | 食品 | 0.253904 | | |
| fruit | 新聞０ | 結果 | 0.669739 | 成果 | 0.144393 | 実 | 0.113671 |
| | 旅行２ | 果物 | 0.571590 | フルーツ | 0.357243 | やつ | 0.071167 |
| gas | 新聞０ | ガス | 0.767677 | ガソリン | 0.181818 | ほら | 0.050505 |
| | 旅行２ | ガソリン | 0.751585 | ガス | 0.194515 | ほら | 0.053901 |
| guide | 新聞０ | 指針 | 0.375734 | 動かす | 0.252446 | ガイド | 0.133072 |
| | 旅行２ | ガイド | 0.663889 | ガイドブック | 0.205271 | 目印 | 0.099063 |
| introduction | 新聞０ | 導入 | 0.406940 | 紹介 | 0.282109 | 採用 | 0.166742 |
| | 旅行２ | 紹介 | 1.000000 | | | | |
| number | 新聞０ | 数 | 0.338364 | 号 | 0.107546 | 量 | 0.082446 |
| | 旅行２ | 番号 | 0.544577 | 号 | 0.308071 | 物 | 0.069416 |
| order | 新聞０ | 目 | 0.608705 | 状態 | 0.130573 | 種類 | 0.077118 |
| | 旅行２ | 注文 | 0.656225 | 目 | 0.219622 | 種類 | 0.077547 |
| prescription | 新聞０ | 規定 | 0.360809 | 命令 | 0.222395 | 指示 | 0.160187 |
| | 旅行２ | 処方箋 | 0.642939 | 処方せん | 0.149954 | 処方 | 0.127133 |
| reservation | 新聞０ | 制限 | 0.556180 | 予約 | 0.345506 | 保留 | 0.098315 |
| | 旅行２ | 予約 | 0.999990 | 制限 | 0.000010 | | |
| straw | 新聞０ | わら | 1.000000 | | | | |
| | 旅行２ | ストロー | 1.000000 | | | | |
| ticket | 新聞０ | メモ | 0.465950 | あてる | 0.193548 | 札 | 0.093190 |
| | 旅行２ | 切符 | 0.623259 | 札 | 0.213306 | あてる | 0.048103 |
| today | 新聞０ | 現在 | 0.766613 | 現代 | 0.149514 | 今日 | 0.065235 |
| | 旅行２ | 今日 | 0.881250 | 本日 | 0.052552 | 現在 | 0.044318 |
| transfer | 新聞０ | 変える | 0.313235 | 移転 | 0.169118 | 移動 | 0.168382 |
| | 旅行２ | 乗り換える | 0.380654 | 乗り換え | 0.311988 | 変える | 0.155436 |

## ５－２：第１近似手法の分析

次に、目的言語の頻度のみを用いる従来手法と、元言語と目的言語の両方を用いる本提案手法の比較を行なって本提案手法の特徴を明らかにする。コーパスの違いによる差異を除くために、以下では、同じ旅行会話コーパスを対象にした手法の違いを比較する。まず本節では第１近似（式（５））の効果を考察する。表２に実験結果の一部を示す。表２にお

いて１行目（モデル０）は目的言語の頻度順を用いた訳語上位候補、２行目（モデル１）は式(５)に従って計算した本方式第１近似の訳語上位候補を表す。３行目（モデル２）は式（８）に従って計算した本方式第２近似の訳語上位候補を表す。表２に挙げた例は、目的言語の頻度のみを用いた場合の第１訳語と、本方式第１近似で計算した第１訳語とが異なっているもので、かつ本方式第２近似まで計算してもその第１訳語が変わらなかった例である。

[表２]　第１近似手法の結果

| 英単語 | モデル | 日本語訳語１位候補 | 翻訳確率 | 日本語訳語２位候補 | 翻訳確率 | 日本語訳語３位候補 | 翻訳確率 |
|---|---|---|---|---|---|---|---|
| back-ground | 0 | 原因 | 0.500000 | 背景 | 0.500000 | | |
| | 1 | 背景 | 0.506708 | 原因 | 0.493292 | | |
| | 2 | 背景 | 0.500021 | 原因 | 0.499979 | | |
| ceremony | 0 | 儀式 | 0.500000 | 式 | 0.500000 | | |
| | 1 | 式 | 0.529685 | 儀式 | 0.470315 | | |
| | 2 | 式 | 0.500117 | 儀式 | 0.499883 | | |
| course | 0 | 道 | 0.356322 | コース | 0.195402 | 方法 | 0.166667 |
| | 1 | コース | 0.374100 | 道 | 0.349878 | 走る | 0.076727 |
| | 2 | コース | 0.374072 | 道 | 0.349912 | 走る | 0.076727 |
| glad | 0 | 楽しい | 0.392523 | うれしい | 0.373832 | 喜んで | 0.102804 |
| | 1 | うれしい | 0.621084 | 楽しい | 0.284181 | 喜んで | 0.057656 |
| | 2 | うれしい | 0.402424 | 楽しい | 0.377506 | 喜んで | 0.098349 |
| please | 0 | すみません | 0.522727 | どうぞ | 0.464646 | どうか | 0.012626 |
| | 1 | どうぞ | 0.401381 | すみません | 0.329041 | どうか | 0.269578 |
| | 2 | どうぞ | 0.401298 | すみません | 0.329083 | どうか | 0.269619 |
| reserve | 0 | 制限 | 0.363636 | 取っておく | 0.363636 | 遠慮 | 0.212121 |
| | 1 | 取っておく | 0.299394 | 遠慮 | 0.238658 | 制限 | 0.182232 |
| | 2 | 取っておく | 0.358543 | 制限 | 0.354748 | 遠慮 | 0.215325 |

元言語（英語）の多くの単語から目的言語（日本語）の同じ訳語への翻訳の合流が起こっているため、目的言語（日本語）の頻度だけを用いる従来手法では、必ずしも妥当な訳が選択されない。本手法を用いると、元言語から目的言語への翻訳確率の合流を考慮して全体がバランスするように計算が行なわれるため、より妥当な訳語が上位に挙がってくることが確認された。

## ５－３：第２近似手法の分析

次に本手法第２近似（式（８））の効果を考察する。表３に実験結果の一部を示す。本節でも、コーパスの違いによる差異を除くために、同じ旅行会話コーパスを対象にした手法

の違いを比較する。表３においても、表２と同様に、モデル０は目的言語の頻度順を用いた訳語上位候補、モデル１は式(５)に従って計算した本方式第１近似の訳語上位候補、モデル２は式（８）に従って計算した本方式第２近似の訳語上位候補を表す。

表３(A）は、モデル０、モデル１、モデル２ですべて訳語が異なり最終的に妥当な結果が得られた例を示す。ここでは参考のため、新聞記事コーパスにおける目的言語の頻度順を用いた訳語上位候補（モデル「新聞」）も併せて示した。表３(B)は、モデル０で正しかった訳語が一旦モデル１で変化し再度モデル２によって復活した例を示す。表３(C)は、モデル０では妥当な訳語でなかったものが一旦モデル１で妥当な訳になり再度モデル２によってモデル０の不適当な訳語が復活してしまった例を示す。

[表３（A）]　モデル０、モデル１、モデル２ですべて訳語が異なる例

| 英単語 | モデル | 日本語訳語１位候補 | 翻訳確率 | 日本語訳語２位候補 | 翻訳確率 | 日本語訳語３位候補 | 翻訳確率 |
|---|---|---|---|---|---|---|---|
| age | 新聞 | 時代 | 0.541295 | 期 | 0.189065 | 世代 | 0.095540 |
| | 0 | 長い間 | 0.285714 | 年齢 | 0.285714 | 期 | 0.142857 |
| | 1 | 期 | 0.389419 | 年齢 | 0.294072 | 熟す | 0.148187 |
| | 2 | 年齢 | 0.285761 | 長い間 | 0.284947 | 期 | 0.143531 |
| turn | 新聞 | 向ける | 0.182431 | 変わる | 0.136730 | 変化 | 0.085409 |
| | 0 | 見つける | 0.253968 | 変わる | 0.253968 | 変える | 0.174603 |
| | 1 | 回る | 0.157626 | 変わる | 0.157320 | 見つける | 0.103695 |
| | 2 | 変わる | 0.249534 | 見つける | 0.248417 | 変える | 0.165575 |

[表３（B）]　モデル０で正しかった訳語がモデル２で復活した例

| 英単語 | モデル | 日本語訳語１位候補 | 翻訳確率 | 日本語訳語２位候補 | 翻訳確率 | 日本語訳語３位候補 | 翻訳確率 |
|---|---|---|---|---|---|---|---|
| need | 0 | 必要 | 0.956522 | 義務 | 0.021739 | 要求 | 0.021739 |
| | 1 | 義務 | 0.509503 | 要求 | 0.490497 | 必要 | 0.000000 |
| | 2 | 必要 | 0.762588 | 義務 | 0.200481 | 要求 | 0.036930 |
| person | 0 | 人 | 0.979167 | からだ | 0.016667 | 身体 | 0.004167 |
| | 1 | からだ | 0.901716 | 身体 | 0.098284 | 人 | 0.000000 |
| | 2 | 人 | 0.919380 | からだ | 0.077228 | 身体 | 0.003391 |
| trip | 0 | 旅行 | 0.726316 | 旅 | 0.136842 | 上げる | 0.063158 |
| | 1 | 過失 | 0.514172 | 旅 | 0.317897 | 体験 | 0.145440 |
| | 2 | 旅行 | 0.690531 | 旅 | 0.140400 | 過失 | 0.079202 |

[表３(C)] モデル１で修正されたモデル０の不正解がモデル２によって復活した例

| 英単語 | モデル | 日本語訳語１位候補 | 翻訳確率 | 日本語訳語２位候補 | 翻訳確率 | 日本語訳語３位候補 | 翻訳確率 |
|---|---|---|---|---|---|---|---|
| blue | ０ | 海 | 0.500000 | 青い | 0.375000 | 空 | 0.125000 |
| | １ | 青い | 0.540495 | 空 | 0.266323 | 海 | 0.193182 |
| | ２ | 海 | 0.497452 | 青い | 0.376374 | 空 | 0.126174 |
| egg | ０ | 人 | 0.959184 | 卵 | 0.040816 | | |
| | １ | 卵 | 1.000000 | 人 | 0.000000 | | |
| | ２ | 人 | 0.916767 | 卵 | 0.083234 | | |

まず表３(A)(B)(C)から次のことが言える。前述したように、目的言語（日本語）の頻度のみを用いた手法（「モデル新聞」「モデル０」）では、その英単語にとってふさわしくない訳語であっても最も出現頻度が高い訳語が第１候補に挙がってしまう。元言語（英語）と目的言語（日本語）の出現頻度のバランスを考慮したモデル１では、日本語訳語の出現頻度が高くても、その頻度が他の英単語の訳語として説明されれば、その訳語の確率すなわち順位が低くなる。他の英単語の訳語として説明されるのは、その日本語訳語を訳語候補に持つ高頻度の英単語が他に存在するような場合が典型例である。turn の「見つける」に対しては find 等、need の「必要」に対しては necessity 等、person の「人」に対しては man 等、blue の「海」に対しては sea 等が存在するため、それら別英単語の出現確率と翻訳確率が高まることによって、当該の英単語に対する訳語候補の順位が変化する。

これを egg の例で説明すると次のようになる。

| 英単語 | 日訳語 | モデル０ | | モデル１ |
|---|---|---|---|---|
| egg (30) --- | 卵(10) | ４．１% | → | 99.99999% |
| --- | 人(235) | ９５．９% | → | 0.00001% |

英単語、日訳語の後ろに記した括弧内の数字はコーパス内における出現頻度である。この例でモデル０すなわち日本語の頻度のみで翻訳確率を計算すると、圧倒的に出現頻度の高い「人」が９５．９%となる。モデル１すなわち英語と日本語の単語の出現確率のバランスを考慮して翻訳確率を計算すると、日本語の訳語、この場合「卵」「人」を訳語にもつ別の英単語が影響してくる。この場合、「卵」を訳語にもつ英単語は egg の他に存在しなかった。一方、「人」を訳語にもつ英単語は、person(出現頻度５４)、man(同３７)、friend(同９４)、party(同３３)など多数存在する。そこで、このような場合には、訳語「卵」に至る翻訳確率のほとんどを「egg」が担い、訳語「人」に至る翻訳確率は、他の英単語からの翻訳確率として分配されることになる。

この例では、そもそも英単語 egg に「人」という訳語が付与されていること自体が辞書

の不備のように感じられるかも知れない。確かにこの訳語はあまり適切な訳語ではないと思われるが、現実の対訳辞書には多数の訳語が付与されており、中にはあまり適当と思われない訳語や特殊な状況でのみ用いられる訳語もつながっているのが実際である。そのような訳語がある場合にも、本方式によれば、不適切な訳語への翻訳確率を低くできる可能性が本実験によって確かめられた。

モデル１はモデル０に比べてこのような特徴をもつが、モデル０、モデル１までの方法では、品詞の制限を考慮していないので、たとえば age　に対する第３候補「熟す」や、trip に対する第３候補の動詞「上げる」のように、品詞の観点で対応しないと思われる訳語候補も上位に挙がってきてしまうという問題は依然として残っている。

係り受け関係にある２単語の同時共起確率を元言語と目的言語とで考慮したモデル２によると、当該の語と係り受け関係にある単語の訳語も計算されるので、より妥当な訳語が上位に挙がってくる。表３(B)に挙げた例では、最も妥当な訳語がモデル１で一旦は変更されてしまっているが、係り受け関係にある単語を考慮することで、再度最も妥当な訳語が第１候補に挙がってきている。また、モデル２では係り受け関係を考慮しているので品詞の制約も加味されるため、上記のような品詞の対応の観点で頻度の低いと思われる訳語候補の順位が下がるという利点がある。

しかしながら、表３(C)では、表３(B)とは逆に、モデル１によって妥当な訳に一旦変化したものが、モデル２によって再度不適当な訳語に戻ってしまっている。表３(B)のような例と表３(C)のような例との差が起こる原因を分析したところ、データの数に依存するらしいことが明らかになった。表３(B)すなわちモデル２によって妥当な訳語が得られる単語は出現頻度が高く、表３(C)すなわちモデル２によって妥当な訳語が得られない単語は出現頻度が低い傾向がある。それを示すのが表４である。

表４：単語の出現頻度とモデル２の効果の関係

| 英単語 | コーパスでの出現頻度 | 係り受け関係の延べ数 | モデル２の効果 |
|---|---|---|---|
| need | 194 | 405 | 有効に作用 |
| trip | 67 | 107 | 有効に作用 |
| person | 54 | 61 | 有効に作用 |
| blue | 14 | 12 | 不適切に作用 |

つまり、係り受け関係の数が少なすぎて、係り受け関係を考慮した式（８）の第２項が有効に作用しなかったものと分析される。同様なことが上述した egg（出現頻度３０）の場合にも起こっている。

このことから、本モデルの当初の式（７）における第１項と第２項の寄与を表すパラメータα、βについて以下の調整が妥当であることが予想される。すなわち、単語の出現頻度、係り受け関係の数が多い場合には、第２項の寄与を大きくするため　β　の値を大きく設定し、単語の出現頻度、係り受け関係の数が少ない場合には、β　の値を小さく設定

することが考えられる。この予想の検証および実際のパラメータ・フィッティングについては今後の課題である。

最後に本方式で適切な訳語の得られなかった例について考察する。表５は、本方式によって第 1 候補に適切な訳語が選ばれなかった例を示す。英単語「like」は「好きだ」の意味の動詞、あるいは「～のような」の意味の前置詞として用いられるが、日本語において「好きだ」「好む」といった単語は実際にはあまり使用されず、意訳した表現が用いられるため、「好きだ」等が上位候補にはいってきていない。「～のような」のような形態素の複合した表現は、本手法では辞書の見出しに挙がっていないため、やはり候補として挙がってくることがない。英単語「make」の場合も同様である。

[表 5]　意訳によって適切な対訳が得られなかった例

| 英単語 | モデル | 日本語訳語１位候補 | 翻訳確率 | 日本語訳語２位候補 | 翻訳確率 | 日本語訳語３位候補 | 翻訳確率 |
|---|---|---|---|---|---|---|---|
| like | 0 | 合う | 0.736842 | 似合う | 0.236842 | 同じく | 0.026316 |
| | 1 | 合う | 0.747401 | 似合う | 0.232072 | 同じく | 0.020527 |
| | 2 | 似合う | 0.399233 | 合う | 0.319833 | 同じく | 0.280934 |
| make | 0 | する | 0.473142 | 行く | 0.194791 | なる | 0.170374 |
| | 1 | する | 0.479179 | 行く | 0.194426 | なる | 0.169985 |
| | 2 | なる | 0.257577 | 持つ | 0.236272 | 行く | 0.220226 |

このような最も基本的な動詞は様々な句の中で用いられる多義語であるが、この種の単語の場合には、本方式の拡張が必要となる。まず、like のように、単語と複合語が対応するような場合にも適応できるよう単語対単語の対訳確率というモデルを拡張する必要がある。また　make のように目的語ごとに訳語が異なるような機能動詞に関しては、式（８）の第２項のように係り受け関係にある単語対の確率の和をとるだけではなく、係り受け関係の単語対ごとに元言語と目的言語で対応させることで、単語の組み合わせに応じた翻訳確率を求めるようにモデルを拡張する必要がある。

## ６：結論、今後の予定

本プロジェクトでは、コンパラブル・コーパスから訳語選択のための対訳知識を抽出する手法を開発した。本手法は、日英それぞれのコーパスの中から、共起関係、係り受け関係にある単語対を大量に抽出し、日英対訳辞書を用いて日英の単語対間の対応をとることで、対訳辞書の対訳候補の中から、そのコーパスが属する分野で確からしい訳語を選択するものである。目的言語における単語の出現頻度を用いる従来手法と比較した結果、従来手法に比べて適切な対訳候補を選択できることがわかった。また単語対単語の確率を計算する第 1 近似よりも、元の言語で係り受け関係にある単語対が相手言語で係り受け関係にある単語対に対応することを考慮した第２近似の方が良好な結果を与えることがわかった。

今回実験に用いた計算式では、単語の出現頻度を考慮した第１近似の項と、係り受け関係にある単語対の出現頻度を考慮した第２近似の項との寄与の度合いを等しく設定した。今後は、単語の出現頻度、係り受けの数の大小によって、第２近似の寄与の度合いを変化させるパラメータ調整を行ないモデルの精緻化を図る予定である。本方式は、あらかじめ与えられた対訳辞書の単語訳語対に対して翻訳確率を付与するモデルであるが、第２近似の寄与の度合いをパラメータ調整することで、与えられた対訳辞書に不適切な訳語が存在している場合にも、本方式によるとそのような訳語への翻訳確率が低くなる可能性が示唆される。この点に関する条件の明確化および実証は今後の重要なテーマである。またもし不適切な訳語に対する翻訳確率を低く押さえることができるなら、そのことを利用して、あらかじめある程度広い範囲の訳語を設定しておき、その中から妥当な訳語を選択してくることが可能となる。このことは本方式が新訳語獲得技術に拡張できる可能性を示している。この意味からも、本方式の精緻化を今後も継続して行なってゆく予定である。

今後は精緻化したモデルを様々な分野のコーパスに適用して分野依存の訳語辞書の作成コストを軽減させることを目指す。本プロジェクトで開発した対訳知識抽出手法およびその改良手法を組み込んで分野適応により翻訳精度を向上させた機械翻訳システムの製品化を数年後の目標とする。

## 謝辞

今回の実験にあたっては、様々な機関の言語資源を利用させていただいた。日本語テキストの解析に際しては、京都大学情報学研究科黒橋研究室で開発された日本語形態素解析システムＪＵＭＡＮ、構文解析システムＫＮＰ、および東京工業大学大学院情報理工学研究科田中研究室・徳永研究室で開発された日本語解析システムＭＳＬＲ Parser を使用した。英語テキストの解析に際しては、東京大学大学院理学系研究科辻井研究室で開発されたＸＨＰＳＧパーサー、およびニューヨーク大学で開発された Apple Pie Parser を使用した。また英日対訳辞書としては、日本電子化辞書株式会社で開発されたＥＤＲ電子化辞書 日英対訳辞書 ver1.5 を利用した。日本語および英語の大規模コンパラブルコーパスとしては、読売新聞社から提供を受けた新聞データを用いた。ここに深い感謝の意を表わす。

[あとがき]

機械翻訳、クロス言語情報検索の精度向上を目的として、コンパラブル・コーパスから対訳知識を抽出する手法の開発を行なった。本手法は、元言語の各単語の出現確率と、元言語の各単語と意味的に対応する目的言語の各単語の出現確率とが一致する、と仮定する対訳確率モデルである。まず日英それぞれのコーパスに対して構文解析を行ない、各言語における単語の出現頻度および係り受け関係にある単語対の出現頻度を抽出する。英語の単語および単語対の出現頻度と英日対訳確率の積の総和が、日本語の単語および単語対の出現頻度を推定するというモデルに基づき、その推定出現確率と、日本語コーパスに基づく実測の出現確率との差分を最小にするように、パラメータである英日対訳確率を定める。新聞記事１年分の英日コーパスおよび英日の旅行会話文各約２万文のコーパスに対して実験を行なって、本手法の有効性を確認した。本手法を導入することで、自然言語処理システムで使用する分野ごとの言語知識の構築コストを大幅に低減でき、様々な分野に対して分野限定の高精度自然言語処理システムの開発が容易になる。

[成果発表、特許等の状況]

特許出願１件：　特願２００１－１４４３３７

「対訳確率付与装置、対訳確率付与方法並びにそのプログラム」

[購入機器一覧]

なし

[付録：参考文献]


[1] 野美山浩； 目的言語の知識を用いた訳語選択とその学習性, 情報処理学会自然言語処理研究会８６－８, １９９１.

[2] 野上宏康, 熊野明, 田中克巳, 天野真家；既存目的言語文書からの訳語の自動学習方式, 情報処理学会第 42 回全国大会, ２Ｃ－６, １９９１.

[3] S.Doi and K.Muraki; Translation Ambiguity Resolution Based on Text Corpora of Source and Target Languages. In Proceedings of COLING92, pp.525-531, 1992.

[4] 北村美穂子, 松本祐治；二言語対訳コーパスからの翻訳知識の自動獲得, 電子情報通信学会 言語理解とコミュニケーション研究会 NLC94-2, １９９４.

[5] H.Kaji and T.Aizono; Extracting Word Correspondences from Bilingual Corpora Based on Word Co-occurrence Information, In Proceedings of COLING96, pp.23-28, 1996.

[6] K.Tanaka, H.Iwasaki; Extraction of Lexical Translations from Non-Aligned Corpora, In Proceedings of COLING96, pp.580-585, 1996.

[7] K.Yamabana S.Kamei, S.Doi, and K.Muraki; A Hybrid Approach to Interactive Machine Translation, In Proceedings of IJCAI-97, pp.324-331, 1997.

[8] K.Yamabana, S.Doi, K.Muraki & S.Kamei; A Language Conversion Front-End for Cross-Language Information Retrieval, CROSS-LANGUAGE INFORMATION RETRIEVAL, G.Grefenstette(Ed.), Kluwer Academic Pub., 1998.

[9] 奥村明俊, 石川開,佐藤研治; コンパラブルコーパスと対訳辞書による日英クロス言語検索, 自然言語処理,VOL.5,NO.4, １９９８.

[10] P.Fung and L.Y. Yee; An IR Approach for Translating New Words from Nonparallel, Comparable Texts, In Proceedings of COLING-ACL98, pp.414-420, 1998.

[11] R. Rapp; Automatic Identification of Word Translations from Unrelated English and German Corpora, In Proceedings of ACL99, pp.519-526, 1999.

電子・情報
99Y補03-110-3

平成１１年度

新エネルギー・産業技術総合開発機構

提案公募事業（産学連携研究開発事業）

研究成果報告書

# 複数言語にまたがる言語知識処理技術の研究
# （翻訳事例ベース生成技術の開発）

平成１３年３月

株式会社日立製作所

平成１１年度　新エネルギー・産業技術総合開発機構　提案公募研究開発事業
（産学連携研究開発事業）研究成果報告書概要

| | |
|---|---|
| 作　成　年　月　日 | 平成１３年3月３１日 |
| 分野／プロジェクトID番号 | 分野：電子・情報分野<br>番号：９９Ｙ補０３－１１０－３ |
| 研　究　機　関　名 | 株式会社日立製作所 |
| 研究代表者部署・役職 | 中央研究所マルチメディアシステム研究部・主任研究員 |
| 研　究　代　表　者　名 | 梶　博行 |
| プ　ロ　ジ　ェ　ク　ト　名 | 複数言語にまたがる言語知識処理技術の研究<br>（翻訳事例ベース生成技術の開発） |
| 研　　究　　期　　間 | 平成１２年　3月１５日　～　平成１３年　3月３１日 |
| 研　究　の　目　的 | 対訳文の間の句（フレーズ）の対応関係を自動抽出する技術を開発する. |
| 成　果　の　要　旨 | 次のステップから構成される句の対応づけ方法を開発した. (i)対訳辞書を参照して対応可能な語のペアを抽出する．(ii)語の対応可能性から句の対応可能性を導出する．(iii)上位の句の対応関係との整合性を考慮することにより句の対応関係の曖昧性を解消する．この方法を日本語および英語のHPSGパーザと結合し,特許明細書とニュース記事のコーパスを用いた評価実験を行なった結果，抽出率が69.1%，正解率が65.8%であった.本技術によって機械翻訳システムの開発コストを低減することできる. |
| キ　ー　ワ　ー　ド | 対訳コーパス，機械翻訳，句の対応づけ，構文解析 |
| 成果発表・特許等の状況 | 学会発表：なし.<br>特許：基本アイデアはプロジェクト提案前の特願平3-315981号により公知．この出願は現在審査中．米国特許5442546号が許可されている. |
| 今　後　の　予　定 | 実用化に向けた改良研究を継続するとともに，自然言語処理応用システムの開発ツールとして利用する. |

Summary of R&D Report for FY 1999 Proposal-Based R&D Program
of New Energy and Industrial Technology Development Organization

| | |
|---|---|
| Date of preparation | March 31, 2001 |
| Field /<br>Project number | Field: Electronics and information technology<br>No. 99Y 03-110-3 |
| Research organization | Hitachi, Ltd. |
| Post of the research coordinator | Senior Researcher, Multimedia Systems Research Department, Central Research Laboratory |
| Name of the research coordinator | Hiroyuki Kaji |
| Title of the project | Multilingual Natural Language Processing Technologies<br>—A Method for Generating a Translation Example Base |
| Duration of the project | March 15, 2000 ～ March 31, 2001 |
| Purpose of the project | Development of a method for automatically identifying the correspondences between phrases of a pair of sentences, one a translation of the other. |
| Summary of the results | The proposed method consists of (i) coupling words by consulting a bilingual lexicon, (ii) coupling phrases based on correspondences between words, and (iii) canceling the correspondences inconsistent with correspondences between upper phrases. The method, combined with Japanese and English HPSG parsers, was evaluated using patent and news texts. It attained 69.1% recall and 65.8% precision. It will reduce the cost of developing machine translation systems. |
| Key Word | Bilingual corpus, Machine translation, Phrasal alignment, Parsing |
| Publication, patents, etc. | Paper: none<br>Patent: Japan Patent Application No. 1991-315981 submitted prior to the project is under examination. US Patent No. 5442546 has been allowed. |
| Future plans | The method will be further improved, and it will be used as a tool for developing natural language processing applications. |

## まえがき

インターネットの普及，企業活動のグローバル化が進むにつれて，複数言語にまたがる言語処理技術の必要性が高まっている．機械翻訳システムの精度向上が要求されるとともに，母国語で検索要求を表現して外国語の情報を検索するクロスランゲージ情報検索システムなど，新しい要求が生まれている．

自然言語処理技術の研究は，１９９０年代に rationalist approach から empiricist approach へとパラダイムのシフトが起こった．実際に使用されている言語のデータ，すなわち対象分野の文書あるいは発話を集めたコーパスから言語に関する知識を抽出し，抽出した知識を言語の解析・変換・生成に利用するというアプローチである．このコーパスベースのアプローチは自然言語処理に新たな局面を開くものと期待されている．

自然言語処理の研究・実用化が多くの企業で進められてきたが，コーパスベースの技術を開発するには，コーパスを研究コミュニティで共有していくことが望まれる．また，コーパスからの知識抽出には最新の言語解析技術を応用していく必要があり，大学等の研究機関との協力が望まれる．このような問題意識をもって，東京工業大学，東京大学，京都大学の３大学と日本電気，日立製作所，富士通，東芝の４企業の共同提案による「複数言語にまたがる言語知識処理技術の研究」を開始した．

コーパスから抽出すべき言語知識にはさまざまな種類があるが，日立製作所は，言語間の構造的な対応に関する知識に焦点をあて，「翻訳事例ベース生成技術の開発」を分担した．対訳文を集めたパラレルコーパスを句（フレーズ）の対応関係が同定されたコーパスに変換する技術である．次世代の翻訳技術として注目されている事例ベース翻訳のキーとなる技術である．

## 研究者名簿

企業分担代表者

梶　博行（株式会社日立製作所 中央研究所 マルチメディアシステム研究部・主任研究員）

研究分担者

森本　康嗣（株式会社日立製作所 中央研究所 マルチメディアシステム研究部・研究員）

研究分担者

小泉　敦子（株式会社日立製作所 中央研究所 マルチメディアシステム研究部・研究員）

# 目　次

# 1 緒 言

本研究の目的は，対訳文の間の句（フレーズ）の対応関係を同定する方法を開発することである．これは，次世代機械翻訳のパラダイムとして有望視されている事例ベース翻訳のキーとなる技術である．事例ベース翻訳は，類似文の翻訳結果を真似て翻訳するという考え方である(Nagao 1984; Sato 1990)．しかし，一つの翻訳事例が丸ごと利用できることはそれほど多くない．事例を部分的に利用すること，一つの文を翻訳するのに複数の事例を組合わせて利用することが必要である．また，翻訳の具体的なプロセスは，翻訳対象文と事例の原文を照合して差分を抽出する処理，事例の訳文中の差分に対応する部分を翻訳対象文に応じて修正する処理から構成される．したがって，翻訳事例としての対訳文は，任意のレベルの句対応が明示されたものでなければならない．

対訳文の構造的な対応の同定に関しては，1990 年頃からいくつかの方法が提案されている（Kaji 1992; Matsumoto 1993; Meyers 1996; Wu 1997; Watanabe 2000）．しかし，いずれも小規模な原理実験を通じて可能性を示すレベルにとどまっている．一つの理由は，基本となる構文解析の技術が未熟であったことである．精度，速度，ロバスト性のいずれの面でも不十分であった．また，対訳文を集めたパラレルコーパスが未整備で，本格的な評価実験を行なうのが困難であった．しかし，構文解析技術は，最近，急速に進歩してきた．また，自然言語処理の研究コミュニティとしてコーパスの整備を進めようという気運が高まってきた．このような状況から，対訳文に対する句の対応づけの研究に本格的に取組むべき時期であると思われる．

本研究は，特に東京大学辻井研究室の協力を得て進めた．辻井研究室で開発された HPSG (Head-driven Phrase Structure Grammar) パーザは高速，高カバレッジであり，また同一の枠組みによる日本語パーザと英語パーザが利用できることなども本研究に好都合であった(Torisawa 1996; Makino 1998; Mitsuishi 1998; Ninomiya 1998; Kanayama 2000)．構文解析技術は辻井研究室から導入し，本研究では，句の対応づけの実用的なアルゴリズムを開発することを目標とした．また，実用化をめざした研究であるので，評価実験には特許明細書とニュース記事の生の対訳文を使用した．

なお，本研究では，当初，事例ベース翻訳への応用に関連して，句の対応がつけられた対訳文の検索方法についても検討する計画をたてた．しかし，パーザと句の対応づけプログラムのインタフェースの開発に工数がかかることが判明したため，中心的課題である句の対応づけに集中して研究を進めた．

以下，第２章で提案方法の基本的な考え方を述べ，第３章でアルゴリズムを詳細化する．第４章で評価実験の結果を報告し，第５章で改良方向の考察および代替アプローチとの比較を行なう．本研究では対象言語対を日本語－英語としたが，提案方法自体は言語対に依存せず，他の言語対にも適用可能である．

# 2　基本アイデア

## 2．1　概　要

提案方法は，図1に示すように，対訳関係にある日本語文と英語文をそれぞれ構文解析したあと，対訳辞書の助けを借りて照合することにより句の対応関係を抽出する．

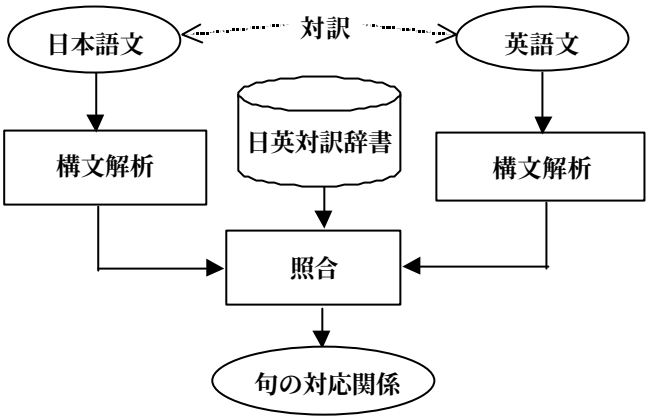



図1　句の対応づけの方法

提案方法を特徴づける考え方は次のとおりである．

(1) 句を内容語が詰められたバッグ（袋）と考えて照合する．すなわち，句の内部の構造は問わない．また機能語も無視する．

(2) 単言語の構文解析では解消できない構文的曖昧性があることを前提とする．言語間の照合処理を通じて解消できる構文的曖昧性は解消する．

(3) 下位の句の対応に基づいて上位の句を対応づけるとともに，上位の句の対応に基づいて下位の句の対応における曖昧性を解消する．

(1)〜(3)の詳細はそれぞれ節を改めて述べる．

## 2．2　内容語のバッグとしての句の照合

### 2．2．1　基本的な考え方

句を内容語の詰まったバッグととらえ,中身の内容語どうしを対応づけることができるバッグが対応すると考える．その理由は次のとおりである．

(1) 自然言語の文は一般に要素合成原理（compositionality principle）に基づいている．すなわち，文の中で句というまとまりを考えることができ，句の意味はそれを構成する句の意味から合成される．

(2) 自然言語の語はモノやコトを表す内容語（名詞，動詞，形容詞など）と文法的な機能を果たす機能語（助詞／前置詞，助動詞など）に分けられる．言語間での内容語の対応は比較的単純である．いっぽう，機能語は言語による違いが大きく，言語間で対応づけるという考え方はなじまない．

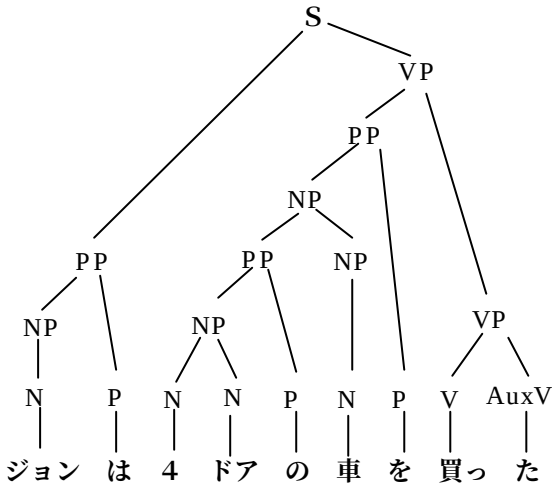
(3) 句構造は言語によって異なり，句構造そのものを照合することは困難である．特に，日本語と英語は語順が異なるため，日本語文とその英訳文は，通常，まったく異なる句構造になる．

対訳例文１の句構造を図２(a)(b)に示す．

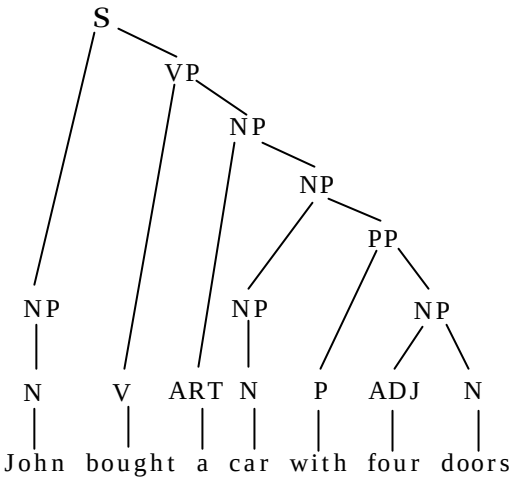
（対訳例文１）

| ジョンは４ドアの車を買った． |
| --- |
| John bought a car with four doors. |

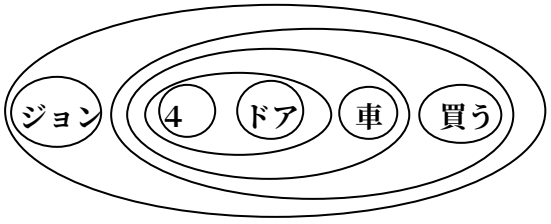
また，句を内容語のバッグととらえた場合の構造を図２(a')(b')に示す．図２の(a)(b)と(a')(b')を比較すると，句構造の言語間照合は困難であるが，句を内容語のバッグととらえることにより句の照合が容易になることが理解できるであろう．



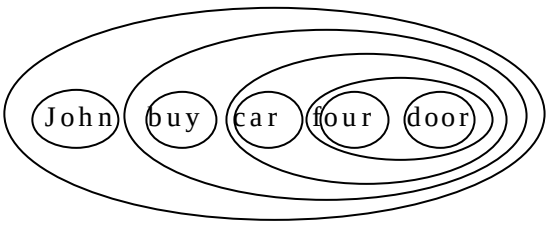

(a) 日本語文の句構造




(b) 英語文の句構造




(a') 日本語文の句の内容語バッグ表現




(b') 英語文の句の内容語バッグ表現

図２　句構造と句の内容語バッグ表現

### ２．２．２　内容語バッグの対応関係から句の対応関係への変換

句を内容語のバッグととらえると，複数の句が同一の内容語バッグに帰着されることがある．例えば，対訳例文１では，三つの句 $(\text{ジョン})_{N}$，$(\text{ジョン})_{NP}$，$(\text{ジョンは})_{PP}$ がいずれも内容語バッグ [ジョン] に帰着され，二つの句 $(\text{John})_{N}$，$(\text{John})_{NP}$ がいずれも内容語バッグ [John] に帰着される（本報告書では，[　] 内に語を列挙することによって内容語のバッグを表わす）．同様に，二つの句 $(\text{４ドア})_{NP}$，$(\text{４ドアの})_{PP}$ がいずれも内容語バッグ [４，ドア] に帰着され，二つの句 $(\text{four doors})_{NP}$，$(\text{with four doors})_{PP}$ がいずれも内容語バッグ [four, door] に帰着される．このような場合，内容語バッグの対応関係を抽出したあと，句の対応関係に変換しなければならない．

内容語バッグの対応から句の対応への変換は次の方法によって行なう．

・日本語文の最小の句と英語文の最小の句を対応づける．

・日本語文の最大の句と英語文の最大の句を対応づける．

・最大でも最小でもない句はどの句とも対応づけない．

この方法によれば，上記の例では，$(\text{ジョン})_{N}$ と $(\text{John})_{N}$，$(\text{ジョンは})_{PP}$ と $(\text{John})_{NP}$ がそれぞれ対応づけられる．また，$(\text{４ドア})_{NP}$ と $(\text{four doors})_{NP}$，$(\text{４ドアの})_{PP}$ と $(\text{with four doors})_{PP}$ がそれぞれ対応づけられる．この対応づけは直観にあったものといえる．

### ２．２．３　対応関係が不明の語を含む対訳文への対処

２．２．１で述べた考え方は，次の条件を満たす場合まったく問題がない．

(a) 対訳の日本語文と英語文の間で内容語の対応関係が一対一である．

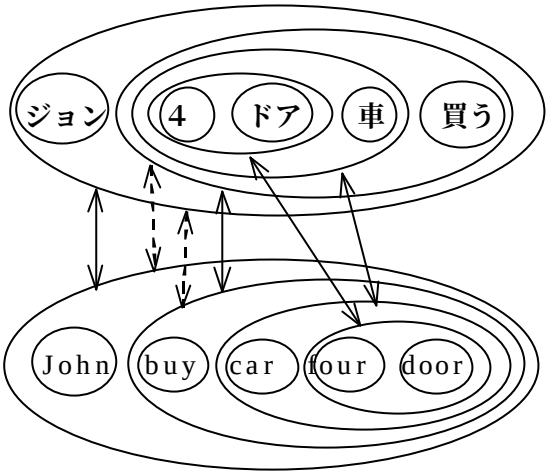
(b) 内容語の対応関係がすべて対訳辞書に登録されている．

しかし，この条件を満たす対訳文はそう多くない．翻訳により作成された対訳コーパスであっても，原文と直接対応しない語を補ったり，原文中の特定の語に対応する語を省略した訳文は多い．また，対訳辞書に 100%のカバレッジを期待することはできない．したがって，内容語バッグの中には対応づけが不可能な語も含まれ得るという前提で，内容語バッグを対応づけるアルゴリズムを設計する必要がある．

ここでは，内容語バッグを対応づける条件を緩和する．すなわち，バッグに含まれる内容語のうち，対訳辞書が示唆する訳語が相手言語の文に存在する語に限定して，バッグ対の対応関係をチェックする．ただし，そうした場合，バッグの対応づけに曖昧性が生じる．これに対処するため，バッグに含まれる内容語（上記対応関係のチェックでは除外された内容語を含む）の数に基づく後処理を行ない，含まれる内容語の数が近いバッグの組を採用する．

図３に例を示す．“ジョン／John”が対訳辞書に未登録であるため，バッグの対応に曖昧性が生じている．破線の矢印で示された対応 [４，ドア，車，買う] ⇔ [John, buy, car, four, door]，[ジョン，４，ドア，車，買う] ⇔ [buy, car, four, door] は正しくない．これらは，バッグに含まれる内容語の数に基づく後処理で棄却され，正しい対応 [４，ドア，車，買う]

⇔ [buy, car, four, door] ，[ジョン，４，ドア，車，買う] ⇔ [John, buy, car, four, door] が残る．




（注）“４／four”，“ドア／door”，“車／car”，
“買う／buy”は対訳辞書に登録されているが，
“ジョン／John”は未登録であるとする．

図３　対応関係が不明の語を含むバッグの対応づけ

## ２．３　構文的曖昧性を含む文の照合

構文的曖昧性は自然言語の文の特徴である．意味を考えないと一意に構造を決定することができない文が多い．句の対応づけはこのことを前提として行なう必要がある．すなわち，構文解析の結果は可能な解をすべて出力し，それらに含まれるすべての句を候補として処理することが必要である。

幸い，構文的曖昧性は言語間で一致するとは限らない．特に，日本語と英語の間ではそうである．このため，言語間での句の対応づけを通じて，単言語では解消できない構文的曖昧性が解消できることがある．例えば，英語でよく問題になる前置詞句の修飾先（PP-attachment）の問題を日本語文との対応づけで解決できることがある．英語文では，前置詞句が直前の名詞句を修飾するのか，その前方の動詞を修飾するのか曖昧である．ところが，日本語文では，名詞句を修飾する場合は連体修飾の句，動詞を修飾する場合は連用修飾の句になり，曖昧性が生じないからである．

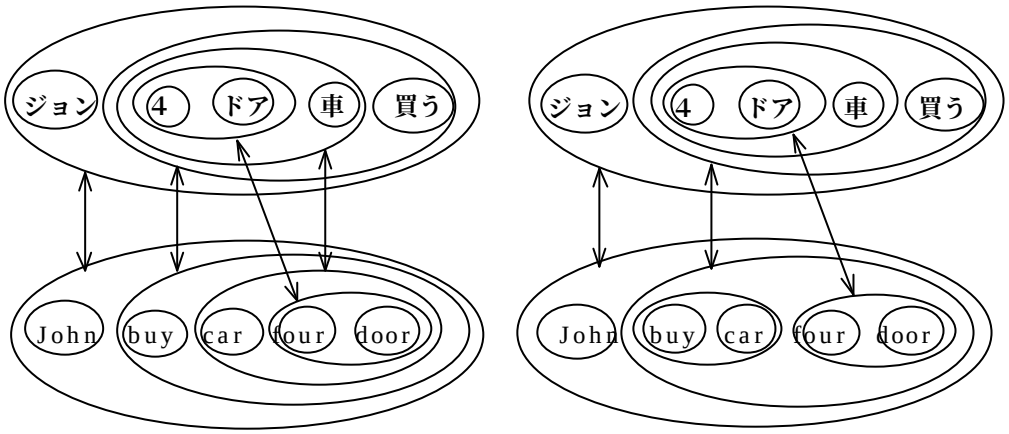
対訳文の句の対応づけを通じて構文的曖昧性が解消される例を図４と図５に示す．図４は対訳例文１（２．２．１参照），図５は対訳例文２に対するもので，ともに英語文に構文的曖昧性がある．

（対訳例文２）

| ジョンは４ドルで車を買った. |
| --- |
| John bought a car with four dollars. |

図４，図５では，図が煩雑になるのを避けるため，句の対応づけ結果のみを示し，語の対応関係は省略した．図４(a)と図５(b)は，英語文の解析が正しい場合で，四つの句すべてが日本語文の句と対応づけられた．図４(b)と図５(a)は，英語文の解析が正しくない場合で，四つの句のうちの一つは日本語文の句に対応づけることができなかった. 構文解析結果に曖昧性が残る場合, 対応づけが可能な句の比率が高い構造を優先的に選択する方法が考えられる.

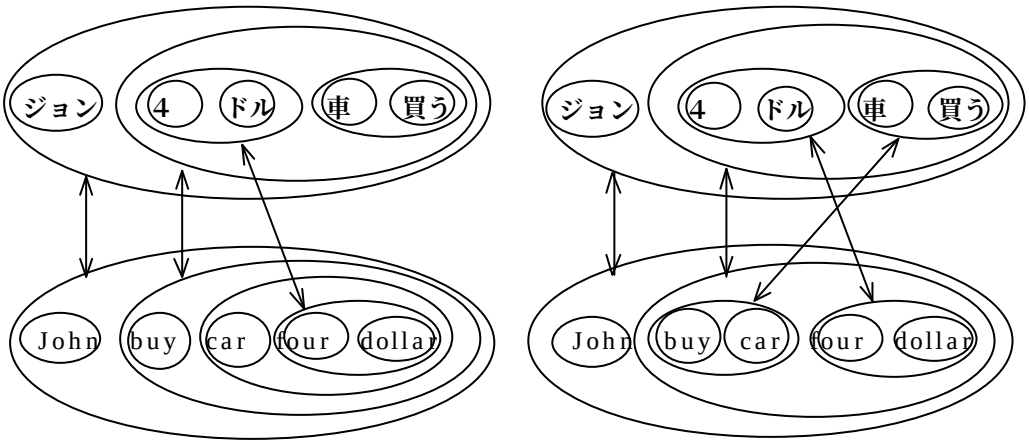
ジョンは４ドアの車を買った. ／John bought a car with four doors.




(a) 英語文の解析が正しい場合　　(b) 英語文の解析が正しくない場合

図４　対訳文の句の対応づけと構文的曖昧性（例１）

ジョンは４ドルで車を買った. ／John bought a car with four dollars.




(a) 英語文の解析が正しくない場合　　(b) 英語文の解析が正しい場合

図５　対訳文の句の対応づけと構文的曖昧性（例２）

## ２．４　ボトムアップ処理とトップダウン処理

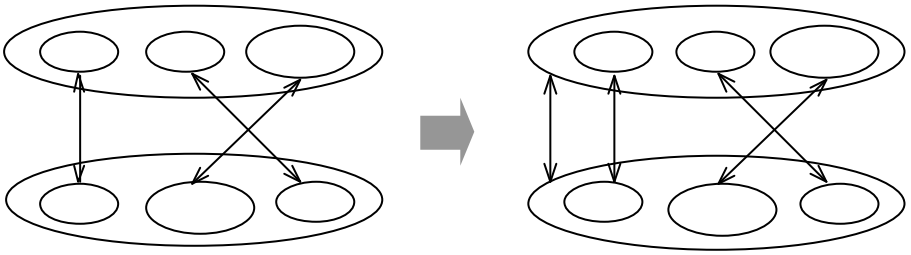
対訳辞書は二つの言語の間で成立し得る対訳語のペアを示唆している．したがって，本当は対応していないのであるが，対訳辞書に登録されている語のペアがたまたま対訳文に含まれることが起こり得る．その場合，語の対応関係に曖昧性が生じる．例えば，対訳辞書に対訳語のペア“車／car”，“車／wheel”，“ホイール／wheel”が含まれているとする．また，対訳の日本語文に“車”と“ホイール”が含まれ，英語文に“car”と“wheel”が含まれるとする．このとき，“車”には“car”が対応するのか“wheel”が対応するのかという曖昧性が生じる．対訳文中での正しい対応関係が“車”と“car”，“ホイール”と“wheel”であるとしても，対訳辞書を参照するだけでは決定できないのである．

同一の語が対訳文中の複数箇所に出現する場合にも，同様な問題が発生する．句の対応づけのために必要な語の対応は，個々の出現箇所すなわちトークンの対応である．一つの語が日本語文中の複数箇所に出現し，対応する語が英語文の複数箇所に出現する場合，対訳辞書を参照するだけで出現箇所ごとの対応を決定することはできない．

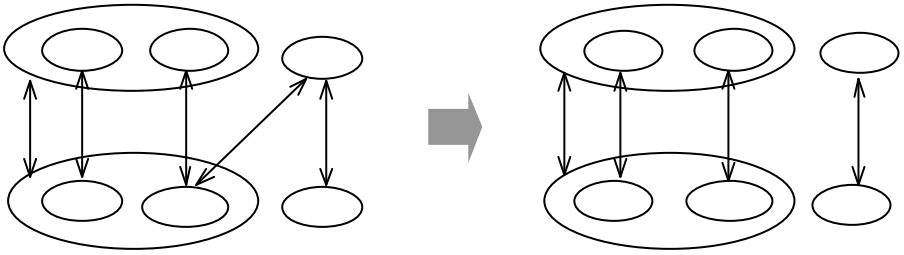
上述の問題を統一的に解決するため，次のようなボトムアップ処理とトップダウン処理を組合わせる．

・ボトムアップ処理：下位の句の対応可能性に基づいて上位の句の対応可能性を抽出する．
・トップダウン処理：上位の句の対応関係に基づいて下位の句の対応の曖昧性を解消する．
　すなわち，上位の句の対応関係と矛盾する下位の句の対応可能性を消去する．

図６にボトムアップ処理とトップダウン処理を図示する．



(a) ボトムアップ処理による対応可能性の抽出



(b) トップダウン処理による対応の曖昧性解消

図６　ボトムアップ処理とトップダウン処理

# 3　アルゴリズム

## 3．1　構文解析結果の表現

対訳の日本語文と英語文それぞれの句構造が句の対応づけ処理の入力となる．2．3で述べたように可能な解すべてが処理の対象であるが,処理効率の面からすべての解がパックされた表現が望ましい．この要求を満たす表現として，ここではCYK（Cocke-Younger-Kasami）解析表を採用する(Aho 1972).

CYK解析表の例を図7(a)に示す．これは，対訳例文1（2．2．1参照）の英語文の解析結果を表わしている．表の列に付けた数字は文中の語の位置を表し，行に付けた数字は語の数を表す．左からi番目，下からj番目の要素A(i,j)には，文中のi番目から(i+j-1)番目の語が構成する句が記される．構文解析の結果，得られる句はすべてこの表に記される．

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | S | | | | | | |
| 6 | | VP | | | | | |
| 5 | | | NP | | | | |
| 4 | | | | NP | | | |
| 3 | | VP | | | PP | | |
| 2 | | | NP | | | NP | |
| 1 | N,NP | V | ART | N,NP | P | ADJ | N |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | John | bought | a | car | with | four | doors |

(a) 解析表

| | | | | | |
|---|---|---|---|---|---|
| 5 | S | | | | |
| 4 | | VP | | | |
| 3 | | | NP;NP | | |
| 2 | | VP | | NP;PP | |
| 1 | N,NP | V | N,NP;NP | ADJ | N |
| | 1 | 2 | 3 | 4 | 5 |
| | John | bought | car | four | doors |

(b) 縮約解析表

図7　CYK解析表とその縮約

次にCYK解析表の縮約について述べる. これは, 句を内容語のバッグととらえること（2．

２．１参照）に相当する．図１１(a)の解析表を縮約した結果を同図(b)に示す．縮約解析表の列に付けた数字は，文中の内容語のみを対象にした語の順序を示し，行に付けた数字は内容語の数を表す．左から i 番目，下から j 番目の要素 B(i,j)には，文中の i 番目から(i+j-1)番目の内容語が詰められたバッグに帰着される句が記される．

解析表を縮約する方法は次のとおりである．

i) 機能語に対応する列の要素の内容語に対応する列への移動

文中，第 i 番目の語が機能語で，その右方で最初の内容語が第 i'番目の語であるとする．各 j (2≦j≦i) について，i+j-1≧i'であるなら A(i',j-(i'-i)) ← A(i',j-(i'-i)) ∪ A(i,j)．i+j-1< i'であるなら何もしない．なお，第 i 番目の語の右方に内容語がないときは何もしない．

ii) 縮約解析表の初期化

全要素を空にする．

iii) 解析表内の内容語に対応する列の要素の縮約解析表への移動

文中，第 i 番目の語が内容語で，第 1 番目から第 i 番目の語のうち機能語が p(i,j)個，第 i 番目から第(i+j-1)番目の語のうち機能語が q(i,j)個であるとき，B(i-p(i,j), j-q(i,j)) ← B(i-p(i,j), j-q(i,j)) ∪ A(i,j)．

## ３．２　句の対応づけ

句を対応づける処理（より正確には，句を表す内容語のバッグを対応づける処理）は次のステップから構成される．

(1) 語対応の候補の抽出
(2) 両立可能なリンクの極大集合の導出
(3) 句対応の候補の抽出
(4) 句対応の曖昧性解消
(5) 最尤対応の選択

以下，各ステップについて説明する．

### ３．２．１　語対応の候補の抽出

日本語文に含まれる語と英語文に含まれる語のペアで，日英対訳辞書に含まれているものをすべて抽出する．抽出されたペアをリンクで結ぶ．

対訳例文３に対し，適当な対訳辞書を仮定した場合の結果を図８に示す．

（対訳例文３）

| |
|---|
| 本発明は情報処理装置の論理回路一般に好適な高速かつ低消費電力の論理回路に関する. |
| The present invention relates to a high speed and low power consumption logic circuit which is generally suitable for logic circuit of a data processor. |

図８では，語の ID を添え字で示し，リンクが結ぶ語の ID をハイフンで結んだものをリ

ンクのＩＤとしている．例えば，“ 3-g ”は“ 論理 3 ”と“ logic g ”を結ぶリンクである．

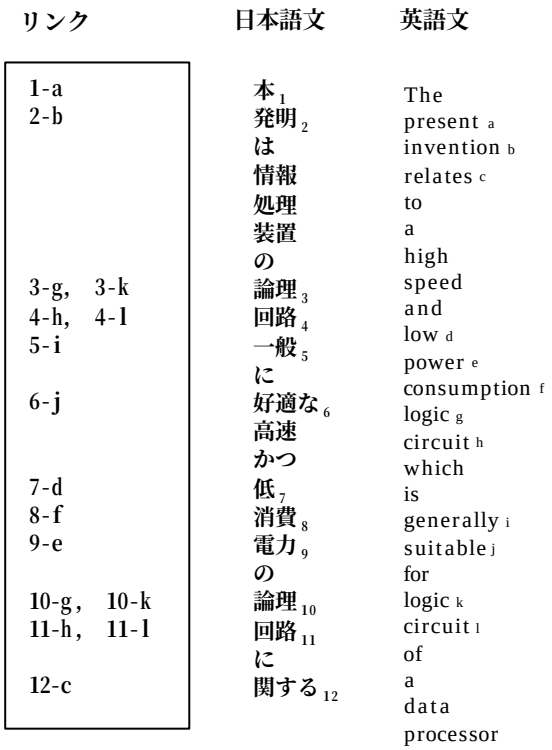



**図８　語対応の候補の抽出例**

### ３．２．２　両立可能なリンクの極大集合の導出

内容語バッグの各々に対して，両立可能なリンクの極大集合 (Maximum compatible link set) の族を導出する．

両立可能なリンクの極大集合とは次のように定義される．同一の語から出るリンクは，どちらか一方しか正しくないので，両立不可能であるという．それ以外のリンクの組は両立可能であるという．一つの内容語バッグに含まれる語から出るリンクの部分集合であって，どの二つのリンクも両立可能であるものを両立可能なリンクの集合という．さらに，同一の内容語バッグに対する他の両立可能なリンクの集合の部分集合ではないとき，両立可能なリンクの極大集合という．以下では両立可能なリンクの極大集合を M.C.L.S.と略記する．

対訳例文３のいくつかの内容語バッグに対して導出される M.C.L.S.の族を以下に示す．

[ 論理 3，回路 4 ]13
　{ {3-g，4-h}，{3-g，4-l}，{3-k，4-h}，{3-k，4-l} }
[ 低 7，消費 8，電力 9 ]14
　{ {7-d，8-f，9-e} }

[ 論理 10，回路 11 ]15

{ {10-g，11-h}，{10-g，11-l}，{10-k，11-h}，{10-k，11-l} }

[ 情報，処理，装置，論理 3，回路 4，一般 5，好適な 6 ]16

{ {3-g，4-h，5-i，6-j}，{3-g，4-l，5-i，6-j}，
{3-k，4-h，5-i，6-j}，{3-k，4-l，5-i，6-j} }

[ 高速，低 7，消費 8，電力 9，論理 10，回路 11 ]17

{ {7-d，8-f，9-e，10-g，11-h }，{7-d，8-f，9-e，10-g，11-l}，
{7-d，8-f，9-e，10-k，11-h }，{7-d，8-f，9-e，10-k，11-l} }

[ low d , power e , consumption f ] m

{ {7-d，8-f，9-e} }

[ logic g , circuit h ] n

{ {3-g，4-h}，{3-g，11-h}，{4-h，10-g}，{10-g，11-h} }

[ logic k , circuit l ] o

{ {3-k，4-l}，{3-k，11-l}，{4-l，10-k}，{10-k，11-l} }

[ high , speed , low d , power e , consumption f , logic g , circuit h ] p

{ {3-g，4-h，7-d，8-f，9-e}，{3-g，7-d，8-f，9-e，11-h}，
{4-h，7-d，8-f，9-e，10-g}，{7-d，8-f，9-e，10-g，11-h} }

[ generally i , suitable j , logic k , circuit l , data , processor ] q

{ {3-k，4-l，5-i，6-j}，{3-k，5-i，6-j，11-l}，
{4-l，5-i，6-j，10-k}，{5-i，6-j，10-k，11-l} }

### ３．２．３　句対応の候補の抽出

共通の M.C.L.S.をもつバッグをリンクで結ぶ処理をボトムアップに行なう．すなわち，日本語の内容語バッグ $B_i$ と英語の内容語バッグ $B_x$ に共通の M.C.L.S.が存在するとき，以下の処理を行なう．

(a) $B_i$ と $B_x$ をリンク $L_{i\text{-}x}$ で結ぶ．

(b) $B_i$ と $B_x$ に共通の M.C.L.S.に“有効”の印をつける．

(c) $B_i$ および $B_x$ の上位の内容語バッグに対する M.C.L.S.のうち，$B_i$ と $B_x$ に共通の M.C.L.S.を包含するものに対して次の処理を行なう．

(i) 当該 M.C.L.S.が $L_{i\text{-}x}$ と両立不可能なリンクを含まなければ，当該 M.C.L.S.の要素として $L_{i\text{-}x}$ を追加する．

(ii) 当該 M.C.L.S.が $L_{i\text{-}x}$ と両立不可能なリンクを含むなら，当該両立不可能なリンクを $L_{i\text{-}x}$ で置き換えた M.C.L.S.を新たに作成し，M.C.L.S.の族に追加する．

３．２．２で例示した内容語バッグについて，上記の処理を行なった結果を以下に示す．ここで，“有効”の印がつけられた M.C.L.S.にはアンダーラインを付した．

[ 論理 3, 回路 4 ]13

{ {3-g, 4-h}, {3-g, 4-l}, {3-k, 4-h}, {3-k, 4-l} }

[ 低 7, 消費 8, 電力 9 ]14

{ {7-d, 8-f, 9-e} }

[ 論理 10, 回路 11 ]15

{ {10-g, 11-h}, {10-g, 11-l}, {10-k, 11-h}, {10-k, 11-l} }

[ 情報, 処理, 装置, 論理 3, 回路 4, 一般 5, 好適な 6 ]16

{ {3-g, 4-h, 5-i, 6-j, 13-n}, {3-g, 4-h, 5-i, 6-j, 13-o},
{3-g, 4-l, 5-i, 6-j, 13-n}, {3-g, 4-l, 5-i, 6-j, 13-o},
{3-k, 4-h, 5-i, 6-j, 13-n}, {3-k, 4-h, 5-i, 6-j, 13-o},
{3-k, 4-l, 5-i, 6-j, 13-n}, {3-k, 4-l, 5-i, 6-j, 13-o} }

[ 高速, 低 7, 消費 8, 電力 9, 論理 10, 回路 11 ]17

{ {7-d, 8-f, 9-e, 10-g, 11-h, 14-m, 15-n}, {7-d, 8-f, 9-e, 10-g, 11-h, 14-m, 15-o},
{7-d, 8-f, 9-e, 10-g, 11-l, 14-m, 15-n}, {7-d, 8-f, 9-e, 10-g, 11-l, 14-m, 15-o},
{7-d, 8-f, 9-e, 10-k, 11-h, 14-m, 15-n}, {7-d, 8-f, 9-e, 10-k, 11-h, 14-m, 15-o},
{7-d, 8-f, 9-e, 10-k, 11-l, 14-m, 15-n}, {7-d, 8-f, 9-e, 10-k, 11-l, 14-m, 15-o} }

[ low d, power e, consumption f ] m

{ {7-d, 8-f, 9-e} }

[ logic g, circuit h ] n

{ {3-g, 4-h}, {3-g, 11-h}, {4-h, 10-g}, {10-g, 11-h} }

[ logic k, circuit l ] o

{ {3-k, 4-l}, {3-k, 11-l}, {4-l, 10-k}, {10-k, 11-l} }

[ high , speed , low d, power e, consumption f, logic g, circuit h ] p

{ {3-g, 4-h, 7-d, 8-f, 9-e, 13-n, 14-m}, {3-g, 4-h, 7-d, 8-f, 9-e, 14-m, 15-n},
{3-g, 7-d, 8-f, 9-e, 11-h, 13-n, 14-m}, {3-g, 7-d, 8-f, 9-e, 11-h, 14-m, 15-n},
{4-h, 7-d, 8-f, 9-e, 10-g, 13-n, 14-m}, {4-h, 7-d, 8-f, 9-e, 10-g, 14-m, 15-n},
{7-d, 8-f, 9-e, 10-g, 11-h, 13-n, 14-m}, {7-d, 8-f, 9-e, 10-g, 11-h, 14-m, 15-n} }

[ generally i , suitable j , logic k, circuit l, data , processor ] q

{ {3-k, 4-l, 5-i, 6-j, 13-o}, {3-k, 4-l, 5-i, 6-j, 15-o},
{3-k, 5-i, 6-j, 11-l, 13-o}, {3-k, 5-i, 6-j, 11-l, 15-o},
{4-l, 5-i, 6-j, 10-k, 13-o}, {4-l, 5-i, 6-j, 10-k, 15-o},
{5-i, 6-j, 10-k, 11-l, 13-o}, {5-i, 6-j, 10-k, 11-l, 15-o} }

### ３．２．４　句対応の曖昧性解消

前処理として，“有効”の印がついていない M.C.L.S.をすべて消去する．そのあと，上位のバッグから出るリンクと矛盾するリンクを消去する処理をトップダウンに行なう．すなわち，少なくとも一つのリンクが出ている内容語バッグ B に関して次の処理を行なう．内容語バッグ B の下位の内容語バッグから出るリンクの各々について，当該リンクが B の“有効”印の付いた M.C.L.S.の少なくとも一つに含まれているかどうかをチェックし，いずれにも含まれていない場合は，当該リンク，および当該リンクを含むすべての M.C.L.S.を消去する．

３．２．３の例に対して上記の処理を行なった結果を以下に示す．ただし，前処理のあとに残った M.C.L.S.をすべて示し，上記の処理で消去されたリンクには二重取り消し線を付した．二重取り消し線が付されたリンクを含む M.C.L.S.は，実際には上記処理によって消去されている．

[ 論理 3，回路 4 ]13
　{ {~~3-g~~，~~4-h~~}，{3-k，4-l} }
[ 低 7，消費 8，電力 9 ]14
　{ {7-d，8-f，9-e} }
[ 論理 10，回路 11 ]15
　{ {10-g，11-h}，{~~10-k~~，~~11-l~~} }
[ 情報，処理，装置，論理 3，回路 4，一般 5，好適な 6 ]16
　{ {3-k，4-l，5-i，6-j，13-o} }
[ 高速，低 7，消費 8，電力 9，論理 10，回路 11 ]17
　{ {7-d，8-f，9-e，10-g，11-h，14-m，15-n} }

[ low d , power e , consumption f ] m
　{ {7-d，8-f，9-e} }
[ logic g , circuit h ] n
　{ {~~3-g~~，~~4-h~~}，{10-g，11-h} }
[ logic k , circuit l ] o
　{ {3-k，4-l}，{~~10-k~~，~~11-l~~} }
[ high , speed , low d , power e , consumption f , logic g , circuit h ] p
　{ {7-d，8-f，9-e，10-g，11-h，14-m，15-n} }
[ generally i , suitable j , logic k , circuit l , data , processor ] q
　{ {3-k，4-l，5-i，6-j，13-o} }

### ３．２．５　最尤対応の選択

内容語バッグ B から出るリンクが複数あるとき，それらのリンクの尤度を計算し，尤度

が最大のリンクを選択する．ここで，日本語文の内容語バッグ $B_i$ と日本語文の内容語バッグ $B_x$ を結ぶリンク $L_{i-x}$ の尤度 $P(L_{i-x})$ は $B_i$ と $B_x$ の語数が近いほど高いと考える．文の長さも考慮に入れた次式で定義する．

$$P(L_{i-x}) = 1 - | |B_i| - |B_x| | / ( n_J + n_E )$$

ここに，$n_J$ は日本語文の内容語数，$n_E$ は英語文の内容語数である．

## ４　評価実験

提案方法をインプリメントし評価実験を行なった．日本語および英語のパーザの出力を句の対応づけプログラムの入力とするため，一つの文に対するすべての解をまとめ，３．１で述べた解析表のデータ構造に変換した．句の対応づけプログラムが参照する対訳辞書としては，日英機械翻訳システムの基本語辞書（日本語の見出し語約５万語）を用いた．評価実験用の対訳文は，特許明細書とニュース記事から選んだ．

### ４．１　句の対応づけ結果の例

句の対応づけ結果の例として，対訳例文４と５に対する結果をそれぞれ図９，図１０に示す．

（対訳例文４）

| |
|---|
| ８は同様にメモリ装置２に印加する外部アドレス信号を示す． |
| 8 denotes an external address signal supplied to the memory device 2. |

（対訳例文５）

| |
|---|
| 感光したレジスト膜１０９は，その後現像液によって図１８のような窓を明け，保護膜１１１をドライエッチングする． |
| The resist film 109 photosensitized by the electron beam is then developed with a developer to form a kind of window as shown in Fig. 18, whereafter the protective film 111 is subjected to dry etching. |

図９，図１０では，対応づけの結果を縮約解析表と同形式の表で示した．日本語文に対する表の要素には，当該内容語バッグに対応づけられた英語の内容語バッグ（の英語文に対する表の要素番号）が記され，英語文に対する表の要素には，当該内容語バッグに対応づけられた日本語の内容語バッグ（の日本語に対する表の要素番号）が記されている．対応の正誤に関する情報も付与した．すなわち，表には次の３とおりの要素番号が記載されている．

- 二重取り消し線も下線も施されていない要素番号－プログラムが抽出した対応で，正解であったもの．
- 二重取り消し線が施されている要素番号－プログラムが抽出した対応で，誤りであったもの．

・下線が施されている要素番号－正しい（抽出すべき）対応であるが，プログラムが抽出できなかったもの.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | (1,9) | | | | | | | | | |
| 9 | | | | | | | | | | |
| 8 | | | (2,8) | | | | | | | |
| 7 | | | (3,7) | | | | | | | |
| 6 | | | | | | | | | | |
| 5 | | | | | | ~~(2,5)~~ | | | | |
| 4 | | | (6,4) | | | ~~(3,4)~~ | | | | |
| 3 | | | (7,3) | | | | (3,3) | | | |
| 2 | | | (7,2) | ~~(8,2)~~ | | | ~~(3,2)~~ | (4,2) | | |
| 1 | (1,1) | | (7,1) | (8,1) | (9,1) | (6,1) | (3,1) | (4,1) | (5,1) | (2,1) |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | **8** | **同様** | **メモリ** | **装置** | **2** | **印加** | **外部** | **アドレス** | **信号** | **示す** |

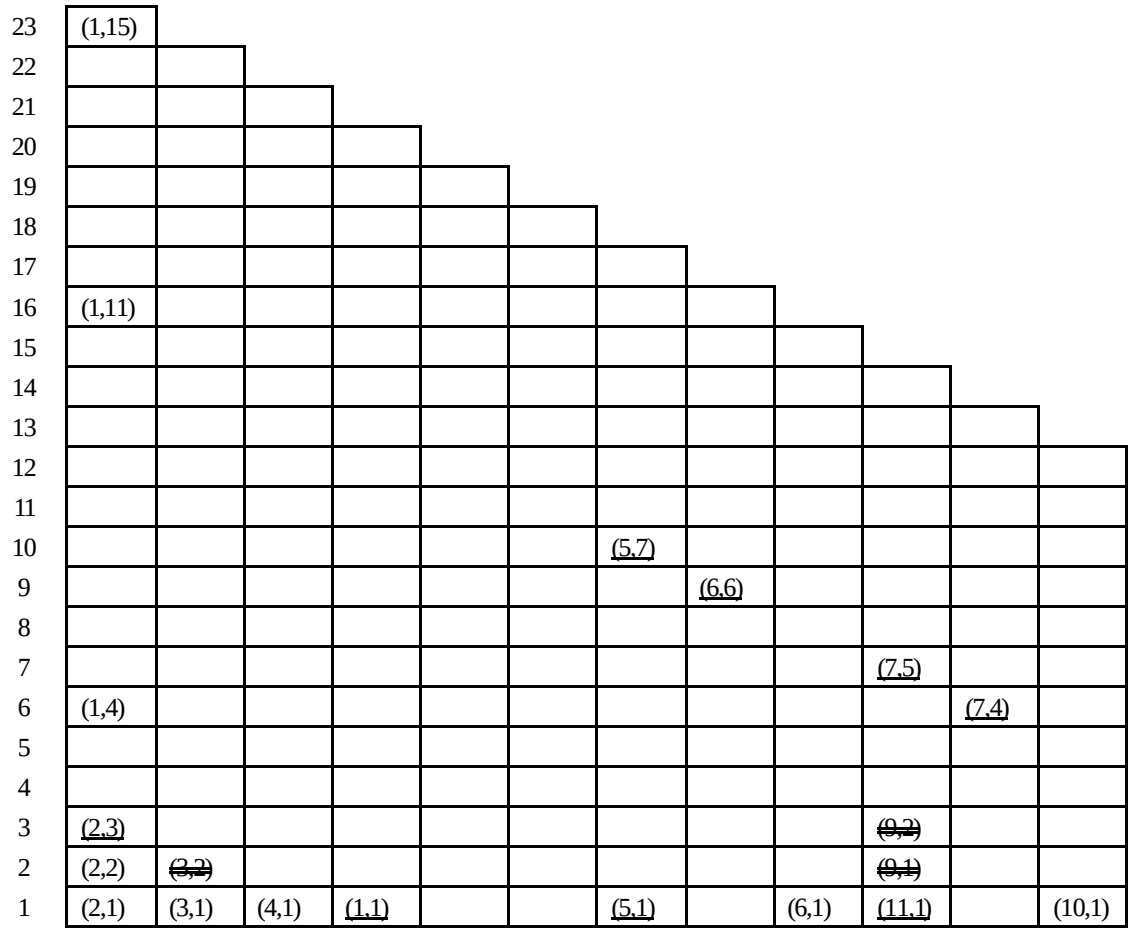
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 9 | (1,10) | | | | | | | | |
| 8 | | (3,8) | | | | | | | |
| 7 | | | (3,7) | | | | | | |
| 6 | | | | | | | | | |
| 5 | | ~~(6,5)~~ | | | | | | | |
| 4 | | | ~~(6,4)~~ | | | (3,4) | | | |
| 3 | | | (7,3) | | | | (3,3) | | |
| 2 | | | ~~(7,2)~~ | (8,2) | | | (3,2) | ~~(4,2)~~ | |
| 1 | (1,1) | (10,1) | (7,1) | (8,1) | (9,1) | (6,1) | (3,1) | (4,1) | (5,1) |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 8 | denote | external | address | signal | supply | memory | device | 2 |

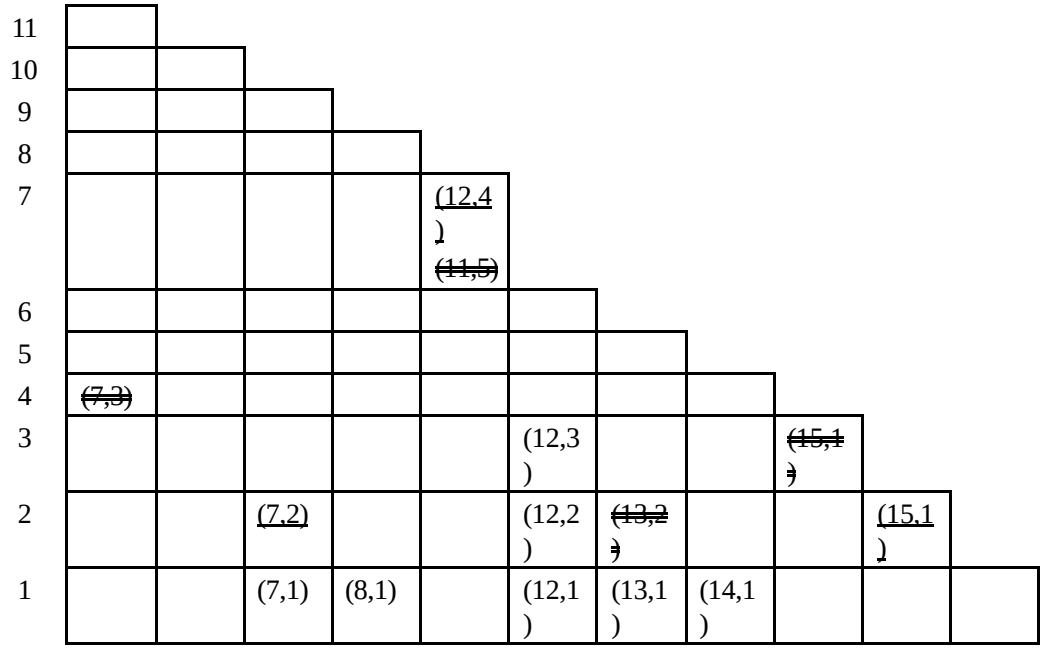
**図９　句の対応づけ結果の例（１）**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 15 | (1,23) | | | | | | | | |
| 14 | | | | | | | | | |
| 13 | | | | | | | | | |
| 12 | | | | | | | | | |
| 11 | (1,16) | | | | | | | | |
| 10 | | | | | | | | | |
| 9 | | | | | | | | | |
| 8 | | | | | | | | | |
| 7 | | | | | (7,10) | | | | |
| 6 | | | | | | (8,9) | | | |
| 5 | | | | | | | (10,7) | | |
| 4 | (1,6) | | | | | | (11,6) | | |
| 3 | | (1,3) | | | | | ~~(13,4)~~ | | |
| 2 | | (1,2) | ~~(2,2)~~ | | | | (15,2) | | ~~(10,3)~~ |
| 1 | (4,1) | (1,1) | (2,1) | (3,1) | (7,1) | (9,1) | (15,1) | (16,1) | ~~(10,2)~~ |
| | **感光** | **レジスト** | **膜** | **１０９** | **その後** | **現像液** | **図** | **１８** | **よう** |

| | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|
| 6 | | | | | | |
| 5 | | ~~(17,7)~~ | | | | |
| 4 | | | (17,7) | | | |
| 3 | | | (18,3) | | | |
| 2 | | | (18,2) | ~~(19,2)~~ | | |
| 1 | (12,1) | (10,1) | (18,1) | (19,1) | (20,1) | (22,2)<br>~~(21,3)~~ |
| | **窓** | **明ける** | **保護** | **膜** | **１１１** | **ドライエッチング** |

**図１０　句の対応づけ結果の例（２）【続く】**


23 (1,15)
16 (1,11)
10 (5,7)
9 (6,6)
7 (7,5)
6 (1,4) (7,4)
3 (2,3) (9,2)
2 (2,2) (3,2) (9,1)
1 (2,1) (3,1) (4,1) (1,1) (5,1) (6,1) (11,1) (10,1)
1 resist
2 film
3 109
4 photosensitize
5 electron
6 beam
7 then
8 develop
9 developer
10 form
11 kind
12 window


7 (12,4) (11,5)
4 (7,3)
3 (12,3) (15,1)
2 (7,2) (12,2) (13,2) (15,1)
1 (7,1) (8,1) (12,1) (13,1) (14,1)
13 show
14 in
15 Fig
16 18
17 where after
18 protective
19 film
20 111
21 subject
22 dry
23 etch

## ４．２　抽出率と正解率

句の対応づけの評価指標としては抽出率 (recall) と正解率 (precision) が考えられる．

・抽出率：正しい対応のうち，プログラムが抽出した対応が占める比率

・正解率：プログラムが抽出した対応のうち，正しい対応が占める比率

実験に用いた対訳例文のうちの 30 対について，人手で抽出した正しい対応と比較して抽出率と正解率を算出した．結果は次のとおりであった．

・抽出率＝69.1%

・正解率＝65.8%

なお，30 対の対訳例文の長さは，日本語文が平均 100 バイト（50 字），英語文が平均 156 バイトであった．また，この 30 文に対する対訳辞書のカバー率，すなわち対訳文に含まれる語の対応のうち対訳辞書に登録されているものの比率は 78%であった．

## ４．３　エラーの分析

対応づけのエラーを分析した．いくつかの要因が複合して生じたエラーも多いが，典型的な要因は以下のとおりである．エラーの例は図９，図１０の例からとった．

(1) 構文解析エラー

(a) 正しい句が抽出されなかったため，抽出もれが生じた．

例（図１０）：

(5,7)「その後現像液によって図１８のような窓を明け」

⇔(7,10)「is then developed with a developer to form a kind of window as shown in Fig. 18」

(b) 誤った句が抽出されたため，不適切な対応が抽出された．

例（図１０）：

(11,5)「明け，保護膜１１１をドライエッチングする」

⇔(17,7)「whereafter the protective film 111 is subjected to dry etching」

(2) 日本語文と英語文に共通の構文的曖昧性

共通の構文的曖昧性があるため，誤った句どうしの対応が抽出された．

例（図９）：

(6,4)「印加する外部アドレス信号」⇔(3,4)「external address signal supplied」

例（図９）：

(7,2)「外部アドレス」⇔(3,2)「external address」

(3) 対訳辞書に未登録の対応の存在／対応する語をもたない語の存在

内容語バッグの対応の曖昧性が生じ，バッグに含まれる語数による最尤対応の選択でも誤りが生じた．

例（図１０）：

(12,4)「保護膜１１１をドライエッチングする」⇔(17,7)「whereafter the protective film 111 is subjected to dry etching」が選択されず，

(11,5)「明け，保護膜１１１をドライエッチングする」⇔(17,7)「whereafter the protective film 111 is subjected to dry etching」が選択された．

（“明ける／form”が対訳辞書に含まれない，日本語文中には“subject”に対応する語が含まれない，などの要因が重なった）

# ５　考　察

## ５．１　改良の方向

実用文を用いた評価実験を通じて提案方法の限界と課題が明らかになった．本節では実用化に向けた改良の方向について考察する．

(1) パーザ，対訳辞書のカバレッジ

提案方法で正解を得るためには，構文解析の解に正しい句構造が含まれていることが前提となる．また，抽出率や正解率は，対訳文に含まれる語の対応のうち，対訳辞書に含まれているものの比率に大きく依存する．そのような意味でパーザ，対訳辞書のカバレッジが重要である．実用文を対象にした場合，この点で問題があることが明らかになった．

パーザについては文法拡充作業を継続していくことが必要である．対訳辞書に関しては，統計的な手法，あるいは基本語対訳辞書を利用したブートストラッピング手法で，コーパスから新しい対訳語のペアを抽出する技術が種々開発されている(Gale 1991; Kupiec 1993; Dagan 1993; 熊野 1994; Fung 1995; Kaji 1996; Kitamura 1996)．これを利用することにより，処理対象のコーパスから対訳語のペアを抽出して対訳辞書に登録した上で，句の対応づけを実行するアプローチが考えられる．

(2) 句対応の尤度の総合的な判定

実用対訳文では，相手言語の文のどの語にも対応しない語がかなり多い．対訳辞書のカバレッジに限界があることと相まって，句の対応を一意に決められないことがかなり多い．３．２．５で述べた単純な尤度が不十分なことが明らかになった．句対応の尤度を判定する方法について抜本的な改良が必要である．

一つは，それぞれの言語の文における句自体の尤度を考慮することが必要である．現在の方法では，パーザが出力する句の候補をすべて対等に扱っている．パーザの内部では，解の信頼度の情報をもっていることが多いが，この情報を引継いで句対応の尤度計算に反映させるべきであろう．

もう一つは，句の構造的な類似性をある程度考えることが必要である．句を内容語のバッグととらえる方法は，句対応の尤度を計算する段階では不適当である．日本語と英語の句構

造はまったく異なるが，句構造における sister の関係を governor-dependent の関係に変換することは容易である．したがって，構造的な類似性を計算することは可能である．

(3) 対象とする句のレベルの限定

複合語から複文まであらゆるレベルの対応を統一的に扱えることが提案方法の一つの特徴である．しかし，今後の改良では対象とする句のレベルを絞ることが重要である．結論を先に述べると，長い複文全体ではなく節（clause）を単位として適用し，また，下位のほうは複合語の内部構造には立ち入らないのがよいと思われる．

長い文に対しては構文解析の精度も悪くなる．また，解の数が組合せ的に増加するので計算量が問題になる．応用面からも，事例ベース翻訳における事例の利用単位は，事例の利用率という面から節程度が適当と思われる．節の対応の抽出はもっと単純でロバストな方法が望ましい．提案方法は節の内部の対応関係を精密に抽出する目的に適している．

複合語については，日本語と英語で構造的曖昧性が共通であり，言語間の対応づけによって解消することは困難である．実は，第４章で述べた評価実験において，エラーのほぼ 3 分の 1 が複合語内部の対応に関するものであった．翻訳システムでは，複合語は対訳辞書で解決するのが実際的であり，(1)で述べた対訳辞書のカバレッジ向上で対処するのがよい．

## ５．２　依存構造に基づく方法との比較

対訳文の構造的な対応づけの代替方法として，依存構造を照合する方法がある (Matsumoto 1993; Watanabe 2000)．本節ではこれと提案方法を比較する．

対訳文の依存構造の例として，対訳例文１（２．２．１参照）と対訳例文６の依存構造をそれぞれ図１１(a)と図１１(b)に示す．

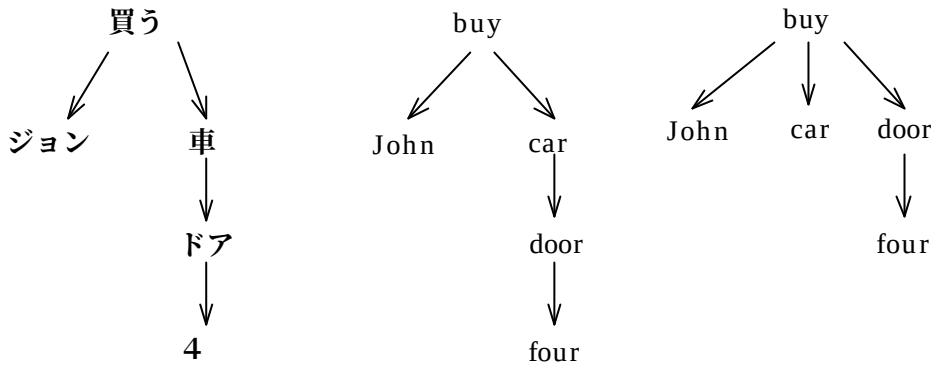
（対訳例文６）

| |
|---|
| メアリは髪が長い． |
| Mary has long hair. |

対訳例文１の英語文は構文的曖昧性があるので，二つの依存構造を示したが，正しい依存構造は日本語文の依存構造と同型になっている．依存構造を照合する方法のほうが提案方法より簡単でよいようにも思われる．句構造に対して句を内容語のバッグと考える工夫をしたが，依存構造を採用すればそのような工夫が不要である．しかし，対訳例文６のように日本語文と英語文の構造が一致しない場合もある．その場合，依存構造の対応づけは困難である．

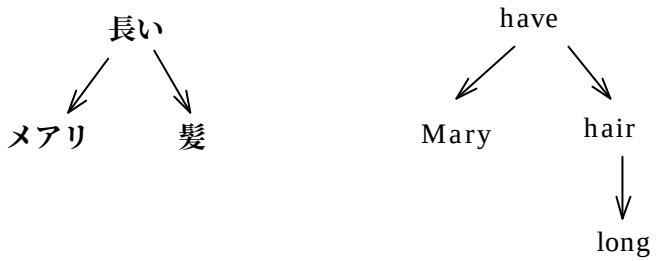
いっぽう，提案方法は，文が要素合成原理に従っているかぎり，言語間の構造の差異を吸収することができる．対訳例文６が提案方法で対応づけられる様子を図１２に示す．依存構造では対応づけが困難な対訳文であるが，内容語のバッグとしての句の対応は素直に抽出することができる．

ジョンは４ドアの車を買った．／John bought a car with four doors.




(a) 言語間で同じ構造になる例

メアリは髪が長い．／Mary has.long hair.




(b) 言語間で構造が異なる例

図１１　対訳文の依存構造

メアリは髪が長い．／Mary has long hair.

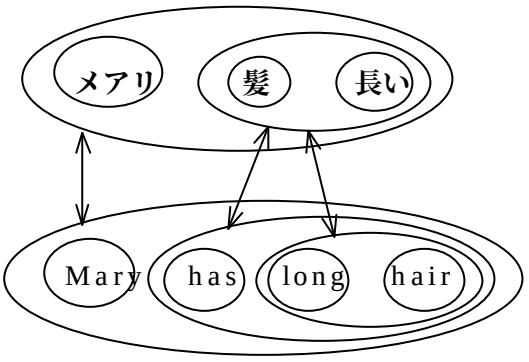



図１２　提案方法による句の対応づけ

句を内容語のバッグととらえることにより,句構造に起因する限界がすべて解決できるわけではない．句の候補は句構造解析の結果によって決まるからである．語順の自由度が高い

日本語には若干問題が生じる. 対訳例文２と日本語文の語順だけが異なる対訳例文７に対する句の対応づけを考える.

（対訳例文７）

| |
|---|
| ジョンは車を４ドルで買った. |
| John bought a car with four dollars. |

図１３に示すように, 英語文から (buy a car)$_{VP}$ が抽出され, これは内容語のバッグ [buy, car] で表わされるが，日本語文から {車, 買う} で表わされる句は抽出されない．句は連続した語の列であって，連続していない語から構成される句はあり得ないからである．この結果，英語文の句 (buy a car)$_{VP}$ は対応する句をもたないことになる.

しかし，対訳文の間の対応という意味では (車を ～ 買った)$_{VP}$ というようなまとまりを考え（不連続であることを「～」で表わす），(buy a car)$_{VP}$ と対応すると考えることも意味があろう．このような要求には，抽出された句の差分どうしを対応づける後処理で対処するのがよいと思われる．図１３の例では，二つの対応関係 (車を４ドルで買う)$_{VP}$ ⇔ (bought a car with four dollars)$_{VP}$ と (４ドルで)$_{PP}$ ⇔ (with four dollars)$_{PP}$ の差分をとることによって，問題の対応関係 (車を ～ 買う)$_{VP}$ ⇔ (bought a car)$_{VP}$ を導出することができる（図１３の斜線部分）.

ジョンは車を４ドルで買った. ／John bought a car with four dollars.

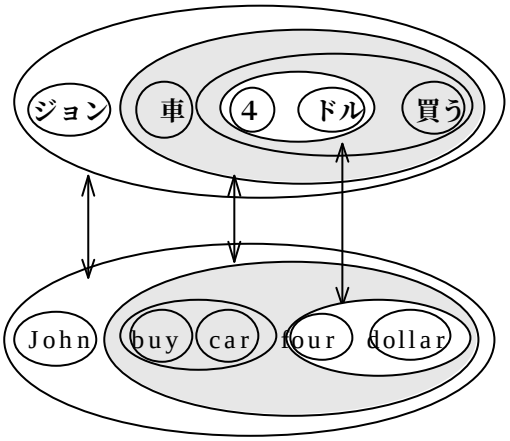



図１３　対応する句の差分の対応づけ

## 6　結　言

対訳の日本語文と英語文を構文解析して照合することにより, 句の対応関係を抽出する方法を開発した．句構造をベースとするが，句を内容語が詰められたバッグととらえることにより，両言語の構造的差異を吸収できるようにした．開発したアルゴリズムは，(i)日英対訳

辞書を参照した語の対応可能性の抽出，(ii)下位の句の対応可能性に基づく上位の句の対応可能性の抽出，(iii)上位の句の対応と矛盾する下位の句の対応可能性の消去，(iv)句を構成する内容語の数に基づく最尤対応の選択，の各ステップから構成される．

この方法をインプリメントし，特許明細書およびニュース記事の日英対訳文から句の対応関係を抽出する実験を行った．この結果，抽出率は 69.1%，正解率は 65.8%であった．主なエラーの要因は，構文解析エラー，両言語に共通の構文的曖昧性，対訳辞書に未登録の対応や対応する語をもたない語の存在であった．

今後の課題は，パーザおよび対訳辞書のカバレッジを向上させること，構文解析結果に対する確信度や構造的な類似性を含めて句対応の尤度を総合的に判定することである．複合語から節（clause）のレベルの句対応が応用面から重要であり，これに対象を絞って改良を進める予定である．

## 7　参考文献


Aho, A. V. and Ullman, J. D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, Vol. 23, pp. 377-403.

Dagan, I., Church, K. W., and Gale, W. A. 1993. Robust bilingual word alignment for machine aided translation. Proceedings of the Workshop on Very Large Corpora, pp. 1-8.

Fung, P. 1995. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pp. 236-243.

Gale, W. A. and Church, K. W. 1991. Identifying word correspondences in parallel texts. Proceedings of the 4th DARPA Speech and Natural Language Workshop, pp. 152-157 .

Kaji, H., Kida, Y., and Morimoto, Y. 1992. Learning translation templates from bilingual text. Proceedings of the 14th International Conference on Computational Linguistics, pp. 672-678.

Kaji, H. and Aizono, T. 1996. Extracting word correspondences from bilingual corpora based on word co-occurrence information. Proceedings of the 16th International Conference on Computational Linguistics, pp. 23-28.

Kanayama, H., Torisawa, K., Mitsuishi, Y., and Tsujii, J.2000. A hybrid Japanese parser with hand-crafted grammar and statistics. Proceedings of the 18th International Conference on Computational Linguistics, pp. 411-417.

Kitamura, M. and Matsumoto, Y. 1996. Automatic extraction of word sequence correspondences in parallel corpora. Proceedings of the 4th Workshop on Very Large Corpora, pp. 79-87.

熊野明, 平川秀樹. 1994. 対訳文書からの機械翻訳専門用語辞書作成，情報処理学会論文誌, Vol. 35, No. 11, pp. 2283-2290.

Kupiec, J. 1993. An algorithm for finding noun phrase correspondences in bilingual œrpora. Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 17-22.

Makino, T., Yoshida, M., Torisawa, K., and Tsujii, J. 1998. LiLFes-Towards a practical HPSG Parser, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, pp. 807-811.

Matsumoto, Y., Ishimoto, H., and Utsuro, T. 1993. Structural matching of parallel texts. Proc. of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 23-30.

Meyers, A., Yangarber, R.. and Grishman, R. 1996. Alignment of shared forests for bilingual corpora. Proceedings of the 16th International Conference on Computational Linguistics, pp. 460-465.

Mitsuishi, Y., Torisawa, K. and Tsujii, J. 1998. HPSG-style underspecified Japanese grammar with wide coverage, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, pp. 876-880.

Nagao, M. 1984. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. Elithorn, A. and R. Bernerji (eds.) Artificial and Human Intelligence, North-Holland, pp.173-180.

Ninomiya, T., Torisawa, K., and Tsujii, J. 1998. An efficient parallel substrate for typed feature structure on shared memory parallel machines, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, pp. 968-974.

Sato, S. and Nagao, M. 1990. Toward Memory-based Translation, Proceedings of the 13th International Conference on Computational Linguistics, pp. 247-252.

Torisawa, K. and Tsujii, J. 1996. Computing phrasal-signs in HPSG prior to parsing, Proceedings of the 16th International Conference on Computational Linguistics, pp. 949-955.

Watanabe, H., Kurohashi, S., and Aramaki, E. 2000. Finding structural correspondences from bilingual parsed corpus for corpus-based translation. Proceedings of the 18th International Conference on Computational Linguistics, pp. 913-918.

Wu, D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, Vol. 23, pp. 377-403.

## あとがき

本研究では,次世代機械翻訳技術として有望視されている事例ベース翻訳のキーとなる技術を開発した．翻訳事例を真似て翻訳するには，事例としての対訳文を単に集めるだけでは不十分で，句レベルの対応を明示した対訳文コーパスを構築しなければならない．従来，この作業は人手で行なわざるを得ず,そのコストが事例ベース翻訳実用化のボトルネックになっていた．本研究は，この問題の解決に向けた一つのステップである．特許文やニュース記事などの実用文における句の対応関係はそれほど単純でなく,句対応づけの完全自動化は不可能である．しかし，本研究の成果をベースとし，さらに改良していくことにより，翻訳事例ベースの作成コストを大きく低減することが期待される．

本研究の効果が及ぶ範囲は事例ベース翻訳に限定されない.従来型の機械翻訳においても,句の対応関係の情報を含む対訳文コーパスは変換ルールの抽出などに利用することができる．また，対訳文コーパスの一方の言語に注目すると，構文情報付きコーパス(bracketed corpus)とみることができる．構文情報付きコーパスから確率付き文法を学習する技術が,最近，急速に進歩しているが，そのトレーニングコーパスとして利用することができる．さらに，格フレーム知識の学習などにも利用することが考えられる．このように，本研究の成果は,自然言語処理の研究基盤となるコーパスの作成ツールとして位置づけることもできる.今後，自然言語処理技術の研究・実用化に活用していきたい.

## 成果発表，特許等の状況

(1) 学会発表

なし.

(2) 特許

本研究の基本アイデアを含む下記出願が,本プロジェクトより前の研究の中で報告者らによってなされている．日本出願は現在審査中であり，米国出願は既に許可されている.

梶博行，木田裕子，森本康嗣，“翻訳テンプレート学習方法,” 特願平 3-315981 号 (1991.11.29).

Hiroyuki Kaji, Yuko Kida and Yasutsugu Morimoto, “System and method for automatically generating translation templates from a pair of bilingual sentences,” US Patent No. 5442546 (Aug 15, 1995).

## 購入機器一覧

なし

## 付　録

なし

電子・情報分野
99Y 補 03-110-4

平成１１年度

新エネルギー・産業技術総合開発機構

提案公募事業（産学連携研究開発事業）

研究成果報告書

# 「複数言語にまたがる言語知識処理技術の研究」
# （コンパラブルコーパスからの対訳文抽出）

平成 13 年 3 月

富士通株式会社

平成11年度 新エネルギー・産業技術総合開発機構 提案公募研究開発事業
（産学連携研究開発事業）研究成果報告書概要

| | |
|---|---|
| 作成年月日 | 平成13年3月31日 |
| 分野／プロジェクトID番号 | 分野：電子・情報<br>番号：99Y補03-110-4 |
| 研究機関名 | 富士通株式会社 |
| 研究代表者部署・役職 | ASPサービス統括部DBサービス部・担当部長 |
| 研究代表者名 | 松井くにお |
| プロジェクト名 | 「複数言語にまたがる言語知識処理技術の研究」<br>（コンパラブルコーパスからの対訳文抽出） |
| 研究期間 | 平成12年3月15日～平成13年3月31日 |
| 研究の目的 | 機械翻訳やクロス言語情報検索等の多言語処理の精度向上には、大量の対訳コーパスが不可欠である。本研究では、実世界に比較的多く存在するコンパラブルコーパスから、対訳コーパスを作成する手法を開発することを目的とする。 |
| 成果の要旨 | コンパラブルコーパスから対訳コーパスを作成するためのシステムを作成した。処理対象となる実際の新聞記事データをプロジェクト用に手配し、この記事を用いてシステムを実行したところ、実用に耐えうるレベルの対訳コーパスが得られることが確認できた。 |
| キーワード | 対訳コーパス、多言語処理 |
| 成果発表・特許等の状況 | |
| 今後の予定 | |

Summary of R&D Report for FY 1999 Proposal-Based R&D Program
of New Energy and Industrial Technology Development Organization

| | |
|---|---|
| Date of preparation | March 31, 2001 |
| Field/<br>Project number | Field: Electronics and information technology<br>No. 99Y03-110-4 |
| Research organization | Fujitsu Limited |
| Post of the research coordinator | Director |
| Name of the research coordinator | Kunio MATSUI |
| Title of the project | Multilingual Natural Language Processing Technologies |
| Duration of the project | March 15, 2000 - March 31, 2001 |
| Purpose of the project | To develop a methodology for producing bilingual corpora from a given comparable corpus. |
| Summary of the results | A system for creating bilingual corpora from comparable corpora has been developed as a result of the research effort. The system has proven to produce an accuracy that is high enough for practical purposes. |
| Keywords | Bilingual corpora, multilingual NLP |
| Publication, patents, etc. | |
| Future plans | |

## まえがき

インターネットの爆発的普及により複数言語にまたがる言語処理応用システム（機械翻訳、情報検索等）の重要性はますます高まってきている。本共同研究では、大量のコーパスから、言語処理技術の高度化に役立つ知識を抽出し、多言語処理応用システムの高度化に直接利用可能な知識を獲得する技術の研究開発を行うことを目的とする。利用する多言語コーパスとして、厳密な翻訳関係にはないが同一の情報内容をもつテキスト（コンパラブル・コーパスと呼ぶ）を用い、そこから翻訳・検索用の知識を抽出する技術を開発する。そして本サブテーマ「コンパラブルコーパスからの対訳文抽出」の目標は、完全な対訳関係にはないコンパラブルコーパスから対訳知識を抽出する技術を開発することである。

現在の機械翻訳システムの持つボトルネックの一つに分野適応能力の限界がある。実際に使用される言葉は、分野によって語彙自体が異なる。さらに、単語間の対訳関係には一般に複数の可能性があり、同じ語であっても、分野ごとに適切な訳語は異なる。適切な訳語選択のためには、例えば、分野ごとに大量の対訳コーパス（例文集）から対訳関係に関する統計量を取得し利用することが考えられる。しかしこのためには、文単位で対訳となった、大量の対訳コーパスを人手で整備する必要があり、多大なコストがかかるため現実的でなかった。そこで本サブテーマでは、まず分野の同一性が保証され、更に人手による翻訳が施された対訳文も一部に混在しているようなコンパラブルコーパスを大量に収集することとした。このようなコンパラブルコーパスの代表的なものとして、新聞記事があげられる。近年新聞記事は多国語でネット上を流れるようになり、多言語処理研究のための素材として多言語新聞記事が用いられるケースが特に欧米を中心に増えてきている。しかしながら日本においては、まだこのような多言語リソースはまれにしか研究利用できないのが実体である。本プロジェクトでは日本におけるコーパス利用の推進を図るとともに、利用可能な多言語リソースの開拓を目指して、日本の主要新聞社と協議を重ねてきた。その結果、本プロジェクトの趣旨に賛同いただいた毎日新聞社および読売新聞社の御協力により、約１０年分の日本語と英語の新聞記事からなるコンパラブルコーパスを作成することが可能となった。

本サブテーマではこの日英コンパラブルコーパスに対して、各言語内でコーパスの言語解析を行ない、さらに言語にまたがった統計処理をほどこすことによって、コンパラブルコーパスから対訳関係にある日・英の文書を同定し、更に対応した日英文書対から対訳文を抽出する技術を開発した。抽出された対訳文からは更に分野に依存した対訳知識を抽出し、その対訳知識を今度は対訳文抽出の高精度化のために利用することにより、段階的に対訳文抽出の精度が上げられる枠組みを築いた。本報告書では、これら対訳文抽出の仕組みと対訳語句抽出技術について報告する。

## 研究者名簿


| | | |
|---|---|---|
| 研究代表者 | 松井 くにお | 富士通株式会社ＤＢサービス部　担当部長 |
| 研究者 | 潮田 明 | 富士通株式会社ＤＢサービス部　担当課長 |
| | 橋本 三奈子 | 富士通株式会社ＤＢサービス部 |
| | 富士 秀 | 富士通株式会社ＤＢサービス部 |
| | 大倉 清司 | 富士通株式会社ＤＢサービス部 |

# 目次

# 1. はじめに

## 1.1. 研究の背景

機械翻訳やクロス言語情報検索等の複数言語を扱う自然言語処理システムは、実用化されているものがあるとはいえ、いずれも十分といえるレベルには達していないというのが実情である。複数の言語をあつかうこれらの技術の水準を一層高める必要がある。自然言語は、人為的に定められた人工物ではないため、その高精度な処理技術の開発にはコーパス（大量の実文例）を収集し、それを詳細に分析・利用することが必須である。特に、複数言語を扱う自然言語処理では、対象となる言語間のコーパスを大量に収集する必要がある。

本共同研究で研究対象としているオントロジーやフレーズの研究では大量の対訳文が必要となるが、このためには大量のコーパスから対訳文を抽出する技術を開発する必要がある。欧米では対訳コーパスから対訳文を抽出する研究が早い段階から行われ、国内においても理想的な対訳コーパスからの抽出技術はある程度確立されている。しかし、実在するコーパスでは、文書内の一部にしか対応関係がなかったり、対応している部分でも厳密な意味での対訳がなかったりするような、いわゆるコンパラブルコーパスが多い。このため、実用システムの開発のためには、コンパラブルコーパスから対訳文抽出を行うこと自体が重要となってくる。本サブテーマでは、コンパラブルコーパスから対訳文を抽出する技術の開発を目標とする。

## 1.2. 用語の定義

**・対訳コーパス**

複数の言語で記述された文書集合によって構成されるコーパスにおいて、全ての文書がその他の言語において対応する対訳文書を持ち、かつ文書内の全ての文がその他の言語において対応する対訳文を持つようなコーパス。「パラレルコーパス」とも言う。例えば、官公庁の報告書とその外国語版などは 1 文単位で厳密に翻訳されている場合が多いが、これらは対訳コーパスと呼ぶことができる。対訳コーパスは、研究用途には扱いやすく便利だが、世の中で実際に作成される翻訳文書全体のなかで対訳コーパスが占める割合は小さい。

**・コンパラブルコーパス**

複数の言語で記述された文書集合によって構成されるコーパスにおいて、複数言語間で内容的に対応する文書があるようなコーパス。ここでいう「内容的に対応する文書」とは、必ずしも完全な対訳ではなく、意訳になっていたり一部の文のみが対応している場合も含む。例えば、日本語新聞の英語版は、外国人にとって興味のある記事のみを選んで訳したり、記事を要約しながら訳すことが多いが、これはコンパラブルコーパスと呼ぶことができる。また、同じ事柄について異なる言語で独立に書かれた文章も、コンパラブルコーパスである。例えば、ある一つのできごとについて異なる言語で独立に書かれた英語の新聞記事と日本語の新聞記事もコンパラブルコーパスの例である。

## 1.3. 委託研究の分担と報告書の構成

当機関の業務内容は、二つの大きな柱から成る。一つは、プロジェクトに参加している各研究機関が共通して使用できる言語資源作成のために、日本語および英語の大規模言語データを収集することである。もう一つは、収集した多言語データからコンパラブルコーパスを作成し、更にそこから対訳コーパスを自動作成するための技術を開発することである。前者で収集した言語データから、後者で用いた自動作成技術を用いて対訳コーパスを作成し、共有化をはかる。

本報告書では、以下のような順序でこの一連の研究業務についての報告を行なう。

**大規模言語データの収集**

主要新聞社数社と交渉を重ね、本委託業務の研究目的に限り参加各研究機関が自由に活用できる体制を構築した。

**対訳コーパス自動作成技術の開発**

本委託業務における上期および下期の研究開発を通じて、コンパラブルコーパスから対訳コーパスを自動作成する技術を完成させた。上期は文書単位の対応付けを行ったが、下期は対応の付いた文書の中から文として対応している部分を抽出する技術を開発した。下期に行った文の対応付けでは、言語をまたいだ文同士の対応度を求めるが、これは文中に含まれる語句の対訳情報およびその対応関係の統計情報を用いることによって算出した。なお、前項で収集した主要新聞社の大規模言語データに対して開発した技術を適用し、その有用性を実証した。

## 2. コンパラブルコーパスと予備調査

当研究機関は、本共同プロジェクトにおける「コーパス共有化作業グループ」の主査として、東京工業大学のグループと協力して、プロジェクト参加機関が共通して使えるようなコンパラブルコーパスの入手に関する手続きを行なった。

本研究で開発する技術は、いかなる複数言語間においても適用可能な技術を目指しているが、実験や評価のしやすさから当面は日本語と英語のデータを対象として実験を行なった。具体的には、本研究期間中に以下のデータの入手を行なった。

- **毎日新聞社の日本語記事および英語記事** 日本語は1990年10月から2000年12月までの記事、英語は1988年7月から2001年3月までの記事
- **読売新聞社の日本語記事** 1990年1月から1999年12月までの記事

本報告書で述べる一連の研究では、毎日新聞の日本語および英語記事をコンパラブルコーパスとして用いて実験を行なった。

### 2.1. コンパラブルコーパスの対訳対応

本研究の対象とする記事がどのような性質を持っているかについて、予備調査を行った。記事を人手で調査したところ、英語記事と日本語記事には、次のような関係があることがわかった。

- **完全対応**（◎）：全ての文が過不足なく対応しているような文書対。言語Aによって記述される文書中の全ての文が、言語Bによって記述される文書中に対訳関係にある文を持っている。
- **部分対応**（○）：文書対において、一部の文が対訳関係となっている。対訳文に関しては、その部分を抽出すれば対訳コーパスとして使える。
- **内容対応**（△）：対訳文は存在しないが、内容的には同等な文書対。対訳コーパスとして使える部分はないが、対訳語句や対訳表現は抽出することができる。

対象記事は、もともとが日本語で書かれたもので、その一部を後で英語に翻訳している。結果として、日本語記事のほうが英語記事よりもかなり多い。また日英で対応する記事間での発行日付を見ると、英語記事は日本語記事より少し後（１～２日内程度）に書かれている。

以下の表では、ある期間中の全英語記事について、対応する日本語記事が存在するか、また存在する場合は上記の分類のいずれに該当するかを表したものである。英語の記事表題の下に、日本語記事表題を並べてある。対応する日本語記事がない場合は、英語記事表題のみとなっている。

## 人手で調査した記事の対応状況

| 日付 | 記事の題名 | 対応 |
|---|---|---|
| 1991/05/05 | Holiday Flea Markets Mushroom | |
| 1991/05/05 | Environment Concern Rises With Age | |
| 1991/05/05 | Long-Term Rail Plan Formulated | |
| 1991/05/05 | Seven Wonders Of Japan--7 Types Of Japanese Ambiguity | |
| 1991/05/05 | School Days (14): Roles | |
| 1991/05/05 | Fashion--A Look At The Tokyo Collections | |
| 1991/05/05 | Child Population Hits Lowest Recorded Level | |
| 1991/05/05 | 子供の数減る、２２１５万３０００人、人口の１７．９％に | ◎ |
| 1991/05/05 | Foreign Workers Promised Better Job Conditions | |
| 1991/05/05 | Housewives Foresee Paying For Waste Disposal | |
| 1991/05/04 | ゴミ回収有料化に主婦の半数以上が賛成――経企庁調査 | ○ |
| 1991/05/05 | Sudden Illness Claims Workers in Their Prime | |
| 1991/05/04 | 働き盛りの死、８人に１人は突然死　男性は女性の３倍――厚生省初調査 | ○ |
| 1991/05/05 | Waseda Hostages Come Home | |
| 1991/05/05 | パキスタンの誘惑事件の早大生３人が帰国　「軽率」「反省」「無謀」と語る | △ |
| 1991/05/05 | Editorial--Taiwan And The Mainland | |
| 1991/05/04 | 中国本土からの亡命者へ優待規定停止――台湾国防部 | △ |
| 1991/05/06 | Editorial--Vietnam Trade In Limelight | |
| 1991/05/04 | ［社説］対越経済協力、半歩踏み出せ | ◎ |
| 1991/05/06 | Film--Last Frankenstein | |
| 1991/05/06 | Bunraku--May Preview | |
| 1991/05/06 | 2 Children Die In Fire Set After Family Tiff | |
| 1991/05/05 | 祖母の放火で、２人の孫が焼死　おかずで息子と口論、自宅に灯油――栃木・日光 | △ |
| 1991/05/06 | Kids' Pocket Money Increase 13 Percent | |
| 1991/05/05 | １年間のお小遣い、２年前の１３％増加　貯蓄も増えた――日本生命調査 | ○ |
| 1991/05/06 | Law Restricts Schools To 'Official' Foreign Students | |
| 1991/05/05 | 交流の芽に法の壁、フィリピン女性の留学申請却下、偽装留学防止の法改正で法務省 | △ |
| 1991/05/06 | Ministry Looking Into Housing Zone Changes | |
| 1991/05/05 | 住宅地に「中高層専用区」、建設推進へ高さの制限を撤廃――建設省方針 | △ |
| 1991/05/06 | Things To Do--Kansai | |
| 1991/05/06 | Tokyo Firm Defaults On 4.5 Mil. Dollars Owed To Soviet | |
| 1991/05/06 | ロケット発射に支障　「早く機材返して」とソ連側――宇宙商法倒産トラブル | △ |
| 1991/05/06 | Things To Do--Kanto | |
| 1991/05/06 | Japan How To--Woodblock Printmaking (5) | |
| 1991/05/06 | Mingei--Itaya-Zaiku | |
| 1991/05/06 | The Metropolitan Library Service In Tokyo. | |
| 1991/05/06 | Tokyo Univ. Hospital's Professionalism Under Fire: Series I | |
| 1991/05/08 | Within My Ken--The Decayama, Toyama | |
| 1991/05/08 | Vanishing--Geisha | |
| 1991/05/08 | Tokyo Univ. Hospital (2): A Mecca For Unofficial Doctors | |
| 1991/05/08 | B'desh Embassy Calls For Aid | |
| 1991/05/06 | 政府発表の犠牲者も１２万５千７百人に――バングラデシュのサイクロン | △ |
| 1991/05/08 | Along The Tokaido--Day Three | |
| 1991/05/08 | Coke Carrier Arrested | |
| 1991/05/08 | ボリビアからコカイン５・７キロ　成田空港で台湾人逮捕――過去最高の押収量 | △ |
| 1991/05/08 | House Member Commits Suicide | |
| 1991/05/08 | End To Labelling Chaos Sought: Guidelines For Vegetables | |
| 1991/05/06 | 「有機」や「無農薬」などの乱立表示にガイドライン設置へ――農水省方針 | ○ |
| 1991/05/08 | Man Detained 16 Years Confirmed Not Guilty | |
| 1991/05/07 | 小野さん、無罪確定　東京高検が上告断念――千葉県松戸市の女性事務員殺害事件 | ○ |
| 1991/05/08 | Editorial--Japan's Promise To Asia | |
| 1991/05/06 | ［社説］厳守しようアジアへの約束――海部首相のＡＳＥＡＮ歴訪終わる | ◎ |
| 1991/05/08 | Free-Lancing Works For Some | |
| 1991/05/08 | Lower House Passes 3 Percent Tax Revision Bill | |
| 1991/05/07 | 消費税見直し法案、あす成立――衆議院大蔵委員会 | △ |
| 1991/05/08 | PM Directs More Aid To ASEAN | |

## 2.2. 記事対応の例

文で対応した部分を網掛けで示した。なお、表題は意訳の場合が多いので判定の対象外としている。

### 2.2.1. 完全対応の例

**英語記事**

Editorial--Vietnam Trade In Limelight
1991/05/06

Vietnam has suddenly begun to draw attention. Planes flying from Bangkok to Ho Chi Minh City are reported to be filled with businessmen. Mitsubishi Oil Co. is about to participate in the development of oil resources off the Vietnam coast, and the opening of a Vietnam route is being planned by Japan Airlines.

Discussion are taking place within the Asian Development Bank for the restart of fin ancing Vietnam which has begun to convert from a planned to a market economy. We believe that the time has come to begin thinking seriously, in a forward looking manner, about economic cooperation with Vietnam.

In the background of the new world attention to Vietnam are that country's adoption of a policy of economic reforms, activation of a market economy, and advance along the path of competition by the introduction of a system of ownership inclusive of private ownership. The results of these measures have not been fully substantiated as yet, but in 1989 the export of 1,1400,000 tons of rice become possible. In fact, Vietnam become the world's third largest rice exporting country next to the United States and Thailand.

It is worthy of note that Vietnam has the possibility of becoming the core of an "Indochina economic bloc" that will be on a par with Thailand which has made a remarkable economic advance.

Five countries, including Indochina's Vietnam, Laos and Cambodia, together with Thailand and Myanmar(formally Burma) is five times the size of Japan. The basins of the Mekong and Irrawady rivers have abundant water resources and a delta zone, and there are mineral resources that could be developed.

Vietnam's move toward a market economy is related to the end of the Cold War and the reduction of aid from the Soviet Union and Eastern Europe.

Meanwhile, Vietnam's relations with the countries of ASEAN are being strengthened at a rapid tempo. President Suharto of Indonesia visited d Vietnam in November last year. As that time Vietnam expressed its desire to join ASEAN.

Although some time might be required before this can take place, the feeling is spreading among the ASEAN countries that "it is not desirable to isolate Vietnam."

In considering relations with Vietnam, the Cambodian Problem cannot be ignored. In January 1979,Vietnamese troops invaded Cambodia. Japan, in protesting this , froze aid to Vietnam. The situation has changed greatly since then. Vietnam withdrew its troops from Cambodia in September1989. From then on the Cambodian problem has been moving toward a peaceful settlement.

Some problems remain. Vietnam has a number of personnel still in Cambodia as technical advisors. The United States id negotiating with Vietnam in regard to American soldiers who became missing in action(MIA) during the Vietnam War. The United states is unlikely to normalize relations with Vietnam until the Cambodian and MIA problems are solved, and it will also be wary about other countries normalizing ties with Vietnam.

**日本語記事**

［社説］対越経済協力、半歩踏み出せ
1991/05/04

経済人の間でベトナムが脚光を浴びている。バンコク発ホーチミン市行きの航空便はビジネスマンたちでいっぱいだそうだ。最近では三菱石油がベトナム沖の油田開発に参加を表明したし、日本航空もベトナム線開設へ動き出している。

アジア開銀総裁の内部でも、計画経済から市場経済の方向へ転換を始めたベトナムに対し、融資を再開しようとする議論が出ている。私たちは対越経済協力を慎重かつ前向きに見直す時期が来たと考える。

ベトナムが世界から注目され始めた背景には、過去数年ドイモイという経済刷新政策をとり、市場メカニズムの活用、私有制を含む所有形態の導入による競争を推進していることがある。その成果はまだ十分あがっているとはいえないが、八九年には一四〇万トンのコメの輸出が可能となり、米国、タイにつぐ世界第三位の輸出国となった。

この国が経済躍進の目覚ましいタイと並んで広義の「インドシナ経済圏」形成の可能性を持っていることも注目に値する。

ベトナム、ラオス、カンボジアの旧仏領インドシナ連邦にタイ、ミャンマー（旧ビルマ）を加えた五カ国は人口一億七〇〇〇万人。その面積は日本の五倍で、メコン、イラワジ両川の流域は豊かな水資源、土壌（デルタゾーン）のほか、開発可能な鉱物資源に恵まれている。

ベトナムが市場経済の方向へ歩み出し世界市場へ向け窓を開こうとしているのは、東西冷戦の終焉とソ連・東欧からの援助の縮小という現実とも関係があろう。

この国と東南アジア諸国連合（ASEAN）との関係強化が急ピッチで進みつつあることにも目を向けるべきだ。昨年十一月にはスハルト・インドネシア大統領も訪越し、この時、ベトナムはＡＳＥＡＮへの加盟を正式に表明した。

加盟に至るまでまだ長い時間が必要だろうが、ＡＳＥＡＮ諸国内部には「ベトナムの孤立化は好ましくない」との判断が広がりつつある。

ベトナムとの関係を考えるとき、カンボジア問題を抜きにするわけにはいかない。この国は一九七九年一月カンボジアに侵攻、これは荒廃して日本は同国向け援助を凍結した。

しかし事態は大きく変わりつつある。ベトナムは八九年九月にカンボジアから軍隊を撤退させた。以来この問題は平和的解決に向け大きく前進し始めている。

もろろん問題も残っている。ベトナムはいまだにカンボジアは技術顧問などの形で多くの要員を残している。米国はベトナム戦争時に行方不明になった米兵（ＭＩＡ）の問題について、この国と交渉中だ。カンボジア問題に加えこの交渉が決着しない限り米国は国交を回復しないだろうし、他の国々が正常化へ動けば神経をとがらすに違いない。

### 2.2.2. 部分対応の例

**英語記事**

Sudden Illness Claims Workers in Their Prime
1991/05/05

One in eight people who passed away in the prime of life died from sudden illnesses, according to a government survey.
In its first ever effort to examine the nation's deaths among working-aged people, the Health and Welfare Ministry concluded that the threat of sudden death is a reality for overworked Japan.
To obtain results, the ministry interviewed surviving family members of 30- to 65-year-old people who died between April and May 1989. The survey was conducted in 10 prefectures which have death rates typical of the national average.
According to the survey, out of 6,529 deaths, 12.2 percent, or 797 of them were sudden deaths. A week or less before they died from illness all of them led normal healthy lives. Taking the average number of deaths that afflict a population of 100,000, 58.5 of the people who died suddenly were men, and 20.5 were women. Thus 2.9 times more men than women died suddenly.
The percentage of deaths in different age groups that resulted from sudden illnesses varied from 11.8percent to 12.3 percent. In a population of 100,000, 10.2 30-year-olds, 24.2 40-year-olds, 61.550-year-olds and 105.1 60- to 65-year-olds died suddenly.
Over 85 percent were attributed to ailments of the brain and heart. Some 34.6 percent were attributed to some sort of stroke, including cerebral infarctions, and subarachnoid hemorrhages, followed by 31.5 percent from cardiac insufficiency, and 19.8 percent from myocardial infarctions, angina pectoris and other such heart conditions stemming from the obstruction of arteries.
Cardiac insufficiencies were the most common cause of death, at 33.1 percent, among men who died on short notice.
At 43.3 percent, circulatory problems in the brain caused more sudden deaths among women than any other ailment.
Some 72 percent of the people who died had been troubled with some kind of irregular physical condition.
High blood pressure was a symptom in 32.4 percent -- 29.9 percent among men and 39.9 percent among women.
When they first became ill, 64.9 percent of them complained of subjective symptoms. The most common symptoms were of fatigue, and sharp pains. However, 16.6 percent who said by the time they discovered the problem it was too late.
The sudden deaths were not without prior notice. Although the survey shows that the people who died within a week after getting ill were fine before that, 40.7 percent felt weak or more susceptible to sickness than usual. The remaining 59.3 percent were healthy.
In particular, 65.9 percent of the people who died from circulatory problems of the brain, and 60.6 percent who died from cardiac insufficiencies, were healthy up to seven days before their deaths. Yet, nearly 70 percent of them complained of some sort of abnormality from the seventh day onward. Over 24 percent of the people who died of circulatory problems in the brain had complained of headaches.
Almost 30 percent of the people who died of cardiac insufficiencies said they were weary, fatigued or felt sharp pains, and 36.7 percent of the people who died of the obstruction of arteries in the heart told other family members of cold sweats, breathing problems and chest pains.
Ailments beset the people most often at around 7 a.m. and 6 p.m. Heart and artery problems struck in the early morning hours. Over 25.3 percent of the people were hit with their ailment while asleep, 13.4 percent were taking a rest or on a break, and 10.3 percent were commuting to work or already there.

**日本語記事**

働き盛りの死、８人に１人は突然死　男性は女性の３倍――厚生省初調査
1991/05/04

働き盛りに命を落とした人の八人に一人は、発病後一週間足らずの「突然死」だったことが、三日付で厚生省のまとめた一九八九年人口動態社会経済面調査の壮年期死亡分析でわかった。特に男性は女性の三倍にものぼり、九割近くが一家の大黒柱を失う悲劇に。直接の死因は脳卒中が三分の一を占めるなど成人病が圧倒的。発病前、七割近くは何らかの体の異常の自覚があり、同省は「シグナルを見落とさず、気付いたらまず病院へ」と警告している。
（社会面に関連記事）
壮年期死亡の分析はこれが初めて。調査は全国から、平均死亡率が典型的な十県を選び、一昨年四、五月の二カ月に亡くなった三十歳以上、六十五歳未満の人について、遺族らから聞き取りをした。
それによると死者六千五百二十九人のうち、死の一週間前には入院はおろか、普段の生活にも何の支障もなかったのに病死した人は七百九十七人で、一二・二%にのぼった。人口十万人当たりの年間の死者数に換算すると、男性は五十八・五人で女性の二十・五人の二・九倍。世代別では一二・三―一一・八%といずれもほぼ同割合だったが、各世代の人口十万人の中の年間死者数にすると、三十歳代の十・二人が六十歳代では百五・一人になり、年齢が上がるにつれて急増する傾向が出た。
死因は、脳こうそくやクモ膜下出血など脳血管疾患（脳卒中）が三四・六%、次いで心不全三一・五%、心筋こうそくや狭心症など虚血性心疾患が一九・八%の順で、脳・心臓障害が八割半を占めた。男性は心不全がトップで三三・一%、女性は脳血管疾患が最多で四三・三%だった。
死ぬ前の健康状態については既往症のあった人が七二・〇%。男女とも高血圧症が多く、おのおの二九・九%、三九・四%、平均三二・四%を占めた。
そして発病前後の状況をみると、自覚症状を訴えていたのは六四・九%。トップは疲労感や全身のけん怠、疼（とう）痛などで全体の二四・〇%。しかし「気付いた時には手遅れ」の人も一六・六%いた。

### 2.2.3. 内容対応の例

**英語記事**

2 Children Die In Fire Set After Family Tiff
1991/05/06

NIKKO, Tochigi -- Two young children were burned to death over the weekend after their grandmother allegedly set fire to the house in a quarrel with her son, it was reported Sunday.
The nine-member family of Tomi Terauchi, 52, was having dinner
Saturday when an argument broke out over the meal.
In the heat of the argument, Terauchi was hit on the head by her fourth son, Masanobu,21, according to Nikko Police Station.
She was so angry that she went straight to the entrance hall where kerosene was kept, splashed it around and then set it alight, police said.
The fire spread quickly and raged through the wooden house, burning adjoining houses.
In the debris, investigators found the charred remains of Shoichi Oda, 2, and Madato Oda, 4, grandson and granddaughter of Mrs. Terauchi.
After reportedly confessing, she was arrested on arson charges.

**日本語記事**

祖母の放火で、２人の孫が焼死　おかずで息子と口論、自宅に灯油――栃木・日光
1991/05/05

　四日午後七時ごろ、栃木県日光市稲荷町、主婦、寺内とみ容疑者（５２）方から出火、木造平屋建て棟割り長屋約一四〇平方メートルを半焼、北隣の同市下鉢石町、自動車整備工、増村優さん（３８）方木造平屋建て住宅約五〇平方メートル▽北西の同所、洋服仕立業、後藤豊一郎さん（７２）方木造モルタル二階建て住宅約一三二平方メートル▽西隣の同所、無職、西谷光男さん（６７）方木造平屋建て住宅約八二平方メートルの三棟を全焼した。寺内容疑者宅の焼け跡から寺内容疑者の二女、尾田美知子さん（１８）の長男翔一ちゃん（2つ）、三男、無職、寺内秀夫さん（２２）の長女美里ちゃん（4つ）の二人が焼死体で見つかった。日光署は同九時半、寺内容疑者を現住建造物放火の疑いで緊急逮捕した。
　同署の調べによると、寺内容疑者は四男正伸さん（２１）と家族九人で夕食中、おかずのことで口論となり、正伸さんに頭を殴られた。寺内容疑者は「かっとなって灯油を玄関にまき、火をつけた」と供述している。

### 2.2.4. 抽出可能な対訳コーパス

対訳コーパスは、上記の完全対応および部分対応の記事対から抽出することができる。完全対応では、どの文とどの文が対応関係にあるかを計算する必要がある。部分対応では、対訳文を認識して抽出する必要がある。以下では、部分対応の記事対から、どのような対訳コーパスが取れるかを人手でシミュレートしたものである。

**人手で作成した対訳コーパスの例**

| | |
|---|---|
| Sudden Illness Claims Workers in Their Prime | 働き盛りの死、８人に１人は突然死　男性は女性の３倍－－厚生省初調査 |
| 1991/05/05 | 1991/05/04 |
| One in eight people who passed away in the prime of life died from sudden illnesses, according to a government survey. | 働き盛りに命を落とした人の八人に一人は、発病後一週間足らずの「突然死」だったことが、三日付で厚生省のまとめた一九八九年人口動態社会経済面調査の壮年期死亡分析でわかった。 |
| To obtain results, the ministry interviewed surviving family members of 30- to 65-year-old people who died between April and May 1989. The survey was conducted in 10 prefectures which have death rates typical of the national average. | 壮年期死亡の分析はこれが初めて。調査は全国から、平均死亡率が典型的な十県を選び、一昨年四、五月の二カ月に亡くなった三十歳以上、六十五歳未満の人について、遺族らから聞き取りをした。 |
| According to the survey, out of 6,529 deaths, 12.2 percent, or 797 of them were sudden deaths. A week or less before they died from illness all of them led normal healthy lives. | それによると死者六千五百二十九人のうち、死の一週間前には入院はおろか、普段の生活にも何の支障もなかったのに病死した人は七百九十七人で、一二・二%にのぼった。 |
| Taking the average number of deaths that afflict a population of 100,000, 58.5 of the people who died suddenly were men, and 20.5 were women. | 人口十万人当たりの年間の死者数に換算すると、男性は五十八・五人で女性の二十・五人の二・九倍。 |
| The percentage of deaths in different age groups that resulted from sudden illnesses varied from 11.8percent to 12.3 percent. In a population of 100,000, 10.2 30-year-olds, 24.2 40-year-olds, 61.550-year-olds and 105.1 60- to 65-year-olds died suddenly. | 世代別では一二・三－一一・八%といずれもほぼ同割合だったが、各世代の人口十万人の中の年間死者数にすると、三十歳代の十・二人が六十歳代では百五・一人になり、年齢が上がるにつれて急増する傾向が出た。 |
| Some 34.6 percent were attributed to some sort of stroke, including cerebral infarctions, and subarachnoid hemorrhages, followed by 31.5 percent from cardiac insufficiency, and 19.8 percent from myocardial infarctions, angina pectoris and other such heart conditions stemming from the obstruction of arteries. | 死因は、脳こうそくやクモ膜下出血など脳血管疾患（脳卒中）が三四・六%、次いで心不全三一・五%、心筋こうそくや狭心症など虚血性心疾患が一九・八%の順で、脳・心臓障害が八割半を占めた。 |
| Cardiac insufficiencies were the most common cause of death, at 33.1 percent, among men who died on short notice. At 43.3 percent, circulatory problems in the brain caused more sudden deaths among women than any other ailment. | 男性は心不全がトップで三三・一%、女性は脳血管疾患が最多で四三・三%だった。 |
| Some 72 percent of the people who died had been troubled with some kind of irregular physical condition. | 死ぬ前の健康状態については既往症のあった人が七二・〇%。 |
| High blood pressure was a symptom in 32.4 percent -- 29.9 percent among men and 39.9 percent among women. | 男女とも高血圧症が多く、おのおの二九・九%、三九・四%、平均三二・四%を占めた。 |
| When they first became ill, 64.9 percent of them complained of subjective symptoms. | そして発病前後の状況をみると、自覚症状を訴えていたのは六四・九%。 |
| The most common symptoms were of fatigue, and sharp pains. | トップは疲労感や全身のけん怠、疼（とう）痛などで全体の二四・〇%。 |
| However, 16.6 percent who said by the time they discovered the problem it was too late. | しかし「気付いた時には手遅れ」の人も一六・六%いた。 |

## 3. 対訳コーパス作成システムの基本構成

本研究の目的は、前節で説明したようなコンパラブルコーパスから、文単位で対応のついた対訳コーパスを作成することである。以下では、本研究を通じて作成した、コンパラブルコーパスから対訳コーパスを作成するシステムの基本的な構成について述べる。

### 3.1. 主な構成要素

今回の研究の対象記事では、英語の記事量より日本語の記事量の方が多い。このため、日本語記事から対応する英語記事を探した方がその逆よりも効率が良いので、英語からの検索システムを作成した。

- **英語キーワード抽出部** ： 検索に先だって、検索対象の英語ファイルからキーワードを抽出する。英語記事を形態素解析で単語・複合語を取り出して原形に戻し、自立語以外を取りさる。
- **日本語キーワード抽出部**： 日本語文書から、検索キーワードとなる文字列を抽出する。形態素解析で単語を切りだし、自立語以外を取り除く。
- **キーワード言語変換部**： 抽出した英語キーワードを日本語キーワードに変換する。変換は、機械翻訳システム用の対訳辞書を用いる。数字等に関しては、独自の変換規則を用いて英語への変換を行なう。このようにして、検索用の日本語キーワードリストを作成する。
- **記事対応付け部**： 英語から変換した日本語キーワードリストを検索キーとして、日本語記事群の各日本語キーワードリストとの比較を行う。検索の結果、一致度の高いキーワードリストに対応する日本語記事から順に並べる。
- **文対応付け部**： 記事対応付けで得られた対応度の高い日英記事対に対して、文単位の対応付け処理を行い、対訳コーパスを得る。

### 3.2. 処理の流れの例

以下に実際の記事に対する処理の例を示す。本研究の対象記事では、英語記事の方が件数が少ないため、英語記事から検索するようにする。以下の例では、ある一つの英語記事に対して、これに対応する日本語記事を検索する処理の流れを示す。

#### 3.2.1. 英語記事からのキーワード抽出

英語記事に対して形態素解析を行い、自立を中心とするキーワードを抽出する。

**入力英語文書**

End To Labelling Chaos Sought: Guidelines For Vegetables
1991/05/08

The Ministry of Forestry and Fisheries has begun to develop a set of guidelines that it hopes will regulate the use of labels on fruits and vegetables, ministry sources said. It is hoped that the set of standards will make it easier for consumers to figure out what processes or chemicals have been used on what they eat, the sources said. The ministry is examining whether a set of guidelines covering the use of labels on processed foods and indicating the calorific content of food sold by restaurants could be enforced by government. Consumers are becoming increasingly concerned with the labels on foods that indicate safety and quality levels, and the results of the ministry's deliberations on the issue are therefore likely to draw a great deal of attention.
A ministry spokesman said there are more than 60 different kinds of labels currently used by the food distribution industry. There are labels attached to unripe fruit to indicate that no agricultural chemicals have been used, as well as labels that indicate fruit is fully ripe, has been picked early in the morning or been grown on a tree.
There are no standards covering the use of these labels however, and therefore often merely serve to confuse consumers. For example, there are labels that say "low amounts of agricultural chemicals used," and others that say "reduced amounts of chemicals used," leaving the consumers ...

**英語キーワードリスト**

| |
|---|
| End |
| Label |
| Chaos |
| Seek |
| Guideline |
| vegetable |
| ministry of forestry and fisheries |
| begin |
| develop |
| guideline |
| hope |
| regulate |
| use |
| label |
| fruits |
| Vegetable |
| ministry |
| source |
| say |
| hope |
| set |
| standard |
| make |
| easy |
| consumer |
| figure |
| processes |
| chemical |
| use |
| eat |
| source |
| say |

### 3.2.2. 英語キーワードから日本語キーワードへの変換

英語記事から抽出した英語キーワードを日本語キーワードに変換する。

まず各英語キーワードをキーとして英日対訳辞書を検索し、日本語表記群を得る。日本語表記群の中から自立語を中心とした日本語キーワードを選択し、日本語キーワード群とする。

**英語キーワードから変換・展開した日本語キーワード**

| 英語キーワード | 変換後日本語キーワード群 | | | | | | |
|---|---|---|---|---|---|---|---|
| end | 終わり | 終 | 終了 | 端 | 了 | | 終わ |
| label | ラベル | レーベル | 分類 | 標号 | レッテル | | |
| chaos | 混乱状態 | 混沌状態 | | | | | |
| seek | 探 | 追い求 | 追求 | シーク | | | |
| guideline | ガイドライン | 指針 | 指導要領 | | | | |
| vegetable | 野菜 | | | | | | |
| ministry of forestry and fisheries | 農林水産省 | 農水省 | | | | | |
| begin | 初め | はじめ | 明け | 開幕 | 開始 | | |
| develop | 開発 | 発生 | 向上 | 発展 | 成長 | 育て | 現像 |
| set | セット | 一組 | 合わせ | 位置 | | | |
| guideline | ガイドライン | 指針 | 指導要領 | | | | |
| hope | 望 | 希望 | 期待 | 心頼み | 望み | | |
| regulate | 調整 | 規則正し | 整え | | | | |
| use | 使用法 | 使用 | 利用 | 利用法 | 利用方法 | 施用 | |
| label | ラベル | レーベル | 分類 | 標号 | レッテル | | |
| fruit | 実 | フルーツ | 果実 | 果物 | 成果 | 結実 | 実を結 |
| vegetable | 野菜 | | | | | | |
| ministry | 省庁 | 本省 | 内閣 | 省 | 聖職 | | |
| source | 出処 | 筋 | 情報筋 | 資料 | 出典 | 情報源 | 発生源 |
| say | 述べ | 示 | 伝え | 申し添え | 前述 | 言行 | |

### 3.2.3. 日本語記事からのキーワード抽出

検索対象の日本語記事群の各記事についてキーワード抽出を行う。

以下の例では、流れを説明するために、入力の英語記事に対して対訳（部分対訳）関係にある典型的な記事一つと、対訳関係にない典型的な記事一つを例としてあげる。実際の処理では、すべての日本語記事に対してキーワード抽出を行う。

日本語キーワード抽出も英語キーワード抽出と同様に、品詞等で検索に有効なものを選別しておく。

**日本語記事（部分対訳）**

| |
|---|
| 「有機」や「無農薬」などの乱立表示にガイドライン設置へ――農水省方針 |
| 1991/05/06 |
| |
| 農水省は、生鮮野菜や果物に付いている「有機」や「無農薬」などの表示について、一定の基準を設けて適正化する方向で検討を始めた。同省筋が五日明らかにしたもので、あわせて加工食品の賞味期間やファミリーレストランに代表される外食産業でのカロリー表示などについても、ガイドラインを設け行政指導できるかどうか検討を進める。食品の安全性や品質をめぐり、消費者の表示に対する関心が高まっているだけに、同省のとりまとめの結果に注目が集まりそうだ。 |
| 青果物の表示では、「無農薬」などのほか、完全に熟したという意味でトマトやミカンなどに使用されている「完熟」、早朝に収穫したことを強調する「朝どり」、木に実っていたことを印象付ける「木成り」など「流通段階で五、六十種類以上の表示が、乱立している」（同省筋）。 |
| しかし、実際には表示の基準がなく、「低農薬」や「減農薬」と付けられた野菜が、どの程度の農薬投入量で生産されたのかは消費者にはわからない仕組みになっていた。 |
| また、公正取引委員会は八八年九月に農産物の表示について調査。農薬が使われているのに「無農薬」と表示したり、化学肥料が投入されているのに「完全有機栽培」と明記している農産物があり、さらに流通段階で業者が勝手に「有機栽培」と表示しているケースがあったとの結果を公表。生産者や流通業者に改善を求めていた。 |

**抽出したキーワード**

| |
|---|
| 有機 |
| 農薬 |
| 乱立 |
| 表示 |
| ガイドライン |
| 設置 |
| 農水省 |
| 方針 |
| 農水省 |
| 生鮮 |
| 野菜 |
| 果物 |
| 付 |
| 有機 |
| 農薬 |
| 表示 |
| 一定 |
| 基準 |
| 設け |
| 適正 |
| 方向 |
| 検討 |
| 始め |
| |

**日本語記事（関連のない内容の記事）**

| |
|---|
| ホテル１０階で火事　床など焦げる　客ら７０人避難し無事　松山 |
| 1991/05/08 |
| |
| 八日午前十一時五十分ごろ、松山市一番町の国際ホテル松山（十階建て、吉永浩三社長）十階の中華レストラン「桃花林」から出火、レストラン内の床と天井計一〇平方メートルを焦がして約十五分後に消えた。 |
| 出火当時、レストランは開店直後で店内に客はいなかった。 |
| 宿泊客はほとんどチェックアウト後で、桃花林の従業員と、他の階のレストランなどにいた客ら計約七十人はエレベーターなどで避難、全員無事。 |
| 国際ホテル松山は客室八十室（百三十二人収容）。 |

**抽出したキーワード**

| |
|---|
| ホテル |
| １０ |
| 火事 |
| 床 |
| 焦げ |
| 客 |
| ７０ |
| 避難 |
| 無事 |
| 松山 |
| 八日 |
| 午前 |
| 十一 |
| 五十 |
| 松山市 |
| 一 |
| 国際 |

### 3.2.4. 日本語キーワードによる検索

英語キーワードリストを日本語に変換したものをキーとして、日本語記事のキーワードリストを検索する。図中、ヒットしたキーワードは網掛けで表している。ここで、対訳関係にある日本語記事から抽出したキーワードリストは、英語キーワードを日本語キーワードに変換したものにヒットする頻度が高いことがわかる。このヒット頻度をもとに対応度の評価値を計算する。

**対訳記事の場合**

| |
|---|
| 有機 |
| 農薬 |
| 乱立 |
| 表示 |
| ガイドライン |
| 設置 |
| 農水省 |
| 方針 |
| 農水省 |
| 生鮮 |
| 野菜 |
| 果物 |
| 付 |
| 有機 |
| 農薬 |
| 表示 |
| 一定 |
| 基準 |
| 設け |
| 適正 |
| 方向 |
| 検討 |
| 始め |
| 同省 |
| 五日 |
| 明 |
| あわせて |
| 加工 |
| 食品 |

**非対訳記事の場合**

| |
|---|
| ホテル |
| １０ |
| 火事 |
| 床 |
| 焦げ |
| 客 |
| ７０ |
| 避難 |
| 無事 |
| 松山 |
| 八日 |
| 午前 |
| 十一 |
| 五十 |
| 松山市 |
| 一 |
| 国際 |
| ホテル |
| 松山 |
| 十 |
| 吉永 |
| 浩三 |
| 社長 |
| 十 |
| 中華 |
| レストラン |
| 桃花林 |
| 出火 |
| レストラン |
| 床 |

### 3.2.5. 評価値計算

上記入力英語記事に類似していると判断された日本語記事を評価値順に並べたもの。ここで、評価値の一番高い１番目の記事が実際の正解。その他の候補に上がっている記事の中には、対訳となるものはなかった。２番目の候補は、内容は入力英語記事と全く異なっているが、たまたまキーワードがヒットして若干の評価値を得た。

実際のシステムでは複数の候補が出力されるが、用途によって、最高得点のものを選ぶ、複数候補をそのまま使う、人手で選ぶなどの使い方ができる。

**対訳記事の候補の例**

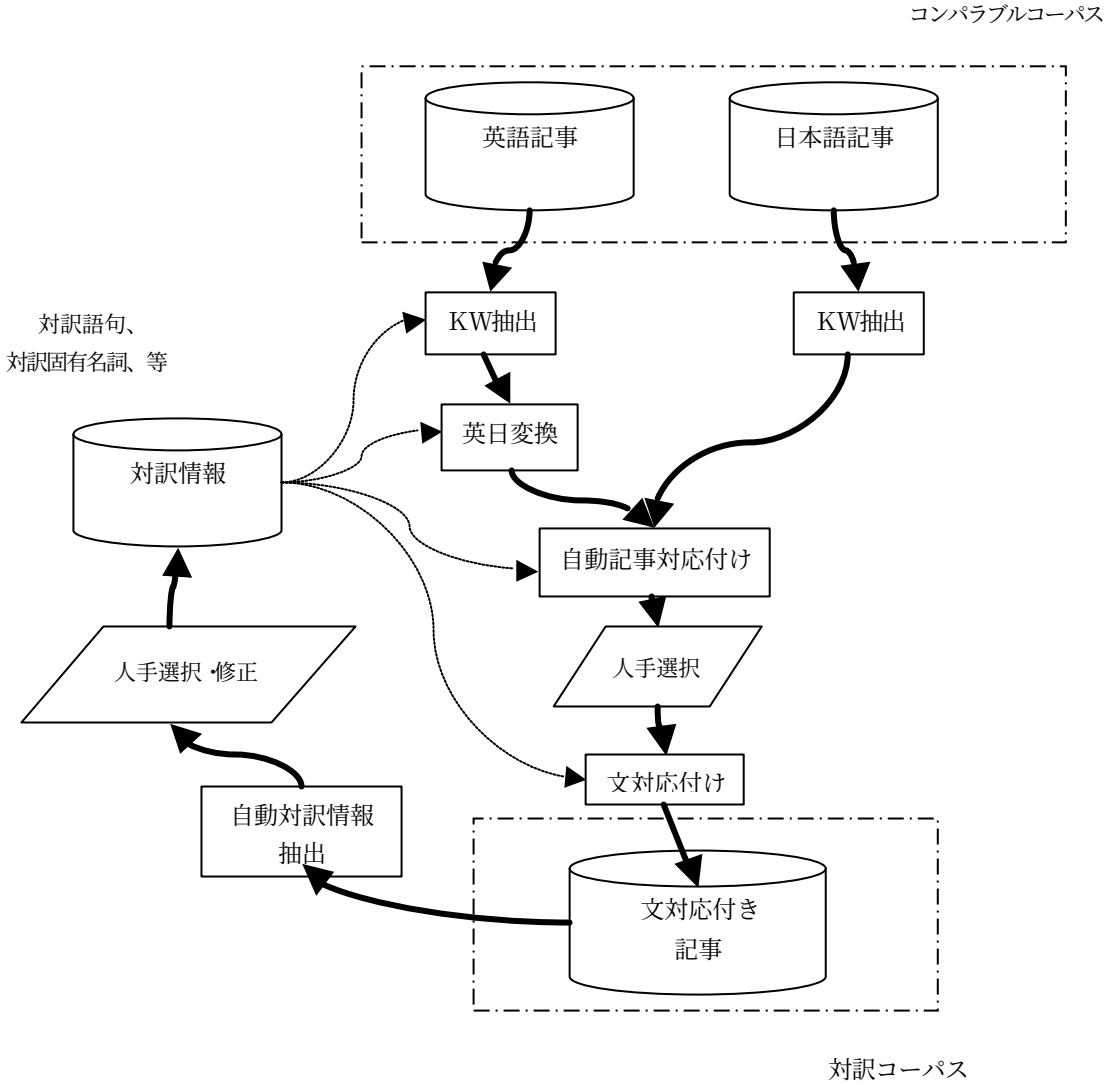
| ファイル名 | 評価値 | 備考 |
|---|---|---|
| 910506M005 | 18.88 | 上記対訳記事 |
| 910507F025 | 3.22 | 上記非対訳記事 |
| 910507F031 | 2.69 | |
| 910507E051 | 2.25 | |

## 4. システムの人手チューニング

基本構成で述べた対訳コーパス作成システムは、一般的なコンパラブルコーパスに対して自動処理できる処理を実現したものである。しかし、実際の対訳コーパス作成では、ある特定の種類のコンパラブルコーパスに対して継続的に処理を行なうことが多く、その対象に合わせて人手でチューニングを行なう必要がある。

図中の人手作業では、システムが自動的に作成した対訳情報の候補から、正しいと思われるものを選択して対訳情報データベースに蓄積することによって、対訳情報が充実してくる。対訳情報がある程度充実すれば、その後はシステムは自動に近い形で運用することができるようになる。チューニング対象となる対訳情報としては、対訳語句、対訳カタカナ語用フィルタ、発信地リストなどがあるが、詳細については、以降の「個々の処理要素」に関する記述を参照されたい。

逆に、蓄積された対訳情報は、形態素解析、キーワード変換、記事対応付け、文対応付けなどの幅広いモジュールで使われる。対訳情報の充実がこれらモジュールの精度向上につながる。



## 5. システムの個々の処理要素

### 5.1. キーワード英日変換部

対応する記事対でも、表記のゆれなどでキーワードがヒットしない場合があり、本来対訳記事として抽出されるべき記事対が取れない場合が多くでてくる。ヒット率を上げるために以下のような処理を行っている。

#### 5.1.1. 表記展開による対応カバー率向上

- **数値表現の展開**：例えば、数字「１」に対して、英語では”1”と表記される場合と”one”と表記される場合がある。また、日本語では、「一」などと漢数字で表現される場合もある。さらには「38億2500万円 → 3.825 billion」というような対応もあり、小数点の移動や単語表記との組み合わせを考慮する必要がある。数値表現は、対応付け処理で重要な役割を果たす場合が多いので､想定される展開規則を作成された対訳コーパスを参考にしながら作成する必要がある。
- **日付の処理**：これは数値表現の一部だが、日付関連でも展開が必要となる。”Jan.”から”January”を展開したり、”’96”から”1996”を展開する必要がある。
- **特殊な別表記の追加**：例えば、「トン」に対して”ton”はあっても”tonne”はないような場合は、後者の綴りで出てくるとヒットしなくなる。このような綴りは、作成した対訳コーパスから自動抽出し、人手で選別して対訳データベースに追加する。
- **単語原形の処理**：これは利用する対訳辞書の文法体系によるが、適切なヒットが起こるような表記処理が必要になる。例えば、システムによっては、”stood”から”stand”も展開しておいたほうが良い場合もある。

#### 5.1.2. 単語クラスによる適合率向上

主に品詞情報を用いて展開を制御する。名詞、動詞、形容詞などは特に変換で重要だが、それ以外の機能語的な語句は、対象記事にあわせて利用するかどうかを調整する。

### 5.2. キーワード重み付け処理部

同じキーワードでも、ヒットしたときに影響の大きなものとそうでもないものが存在する。どんな文脈でも多く出現するキーワードは、ヒットしてもあまり価値がない。反対に、あまり使われないような単語がたまたま出現したような場合には、その単語が対応付けの決め手になる可能性が高い。このことを加味して、以下のような単純な重み付けを行った。
あるキーワードに対する加重頻度は次のように計算する。

$$tf \times \log( K / n )$$

・tf（ターム頻度）： キーワードがある検索対象文書に対してヒットした件数
・K（定数）
・n（タームの全体頻度）：キーワードが検索対象文書全体中で出現する頻度

評価値計算でもう一つ計算に入れなければならないのが、記事全体の大きさである。もともとの記事が長ければ、その中に含まれるキーワードの数が多い可能性が高く、キーワードが全体的にヒットしやすくなる。これも勘案するには、次のような計算で評価値を出す。

ある日本語記事とある検索対象英語文書の文書対について評価値を求めるには、日本語記事から得られた全てのキーワードについて加重頻度を求めてその総和を計算し、検索対象ファイルのファイル長で割る。これを、この文書対の評価値とし、評価値の最も高くなる検索対象文書を出力とする。

### 5.3. 対訳固有名詞（発信地）処理部

今回対象としている記事データでは、記事の発信地情報が安定して得られないが、他の記事ではこの情報が対応付けの際に大きな手がかりとなることが多い。このため、システムでは、記事発信地の対訳情報を蓄積して、これを対応付けの際の候補絞込みのために使っている。以下の２つの方向で、候補数削減を狙っている。

・対応する英語記事のない日本語候補数を削除する
・各日本語記事に対応する英語候補の数を減らす

処理には、発信地の対訳一覧表を事前に用意する。新規の検索対象記事を扱うには、発信地情報を以下のように利用する。なお、日本語・英語の方向は、記事に合わせて適宜変える。

- **未登録発信地の抽出：** 検索対象期間の日本語記事群から発信地を全て抽出し、各々が発信地対訳一覧表にすでに登録されているかどうかをチェックする。未登録の日本語発信地を抜き出す。
- **発信地対訳一覧表への追加：** 未登録の日本語発信地に関しては、一覧表に追加し、その対応する英語発信地を記事中より検索して付加する。なお、英語対訳は必ずしも記事中に存在するとは限らない。
- **対訳検索の実行：** 更新された発信地一覧を用いて対応付けを実行する

多くの記事データでは、記事中の発信地はおおよその書式が決まっているが、個々の記者が手作業で付与するものであり、書式に揺れがある（「ウィーン」と「ウイーン」など）。この揺れを吸収しながら、発信地情報を利用する枠組が必要となる。また、発信地の対訳データを作成する必要がある。

発信地は、例えば「ニューヨーク」、「東京」、「ワシントン」などの大都市が頻度的に圧倒的に多く、これら高頻度のものを網羅しておけば大方の記事には対応できる。しかし、低頻度でも、新規に現れるような発信地は検索結果の絞り込みにおける効果が高いため、これらも事前に登録することは重要である。

以下は、対訳発信地情報の一例である。

**発信地情報の例**

| 英語発信地 | 日本語発信地 |
|---|---|
| Atlanta | アトランタ |
| Amsterdam | アムステルダム |
| Ankara | アンカラ |
| Vienna | ウィーン |
| Westport | ウエストポート |
| Jerusalem | エルサレム |
| Oslo | オスロ |
| Ottawa | オタワ |
| Cairo | カイロ |
| Caracas | カラカス |
| Calgary | カルガリー |
| Canberra | キャンベラ |
| Kuala Lumpur | クアラルンプール |
| Cleveland | クリーブランド |
| Santiago | サンチアゴ |
| Santiago | サンティアゴ |
| San Diego | サンディエゴ |
| San Jose | サンノゼ |
| Sao Paulo | サンパウロ |
| San Francisco | サンフランシスコ |

### 5.4. 文対応重み付け処理部

記事対応付けにおいて対応精度を向上させるには、文単位の重み付けを行うことが有効である。対応度の高い記事対では、単に全体的なキーワードの対応数が多いだけではなく、同じような位置に対応するキーワード対が現れる。この特長を利用して記事へのチューニングを行った。

#### 5.4.1. 対訳関係にある記事対の例

この重み付けのチューニングを行うために、対象記事における文対応を人手で調査した。この調査結果に合うような関数を作成すれば、対応付けの精度を向上させることができる。

なお以下の表では、縦軸が日本語文番号を表し、横軸が英語記事番号を表す。また、表中の各数字は文対の評価値を表す。また、ボールド体の数字は人手による正解対応の位置を表す。

**対訳関係にある記事対の例**（3件）

| J/E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **1** | | 1 | | 0 | | 1 | | | | | | | | | 0 | | | | |
| 1 | 4 | | 1 | 0 | | | 0 | | | | | | | | | | | | 1 | |
| 2 | 1 | | **5** | | | 0 | 1 | 1 | | 1 | 2 | | 0 | 0 | | 0 | 1 | | | 1 |
| 3 | | | 1 | **1** | | 0 | | 0 | 0 | | | | 0 | | | | | | | |
| 4 | | | | | | | | | | | | 0 | | | 0 | | | | | |
| 5 | 1 | 1 | 1 | 0 | | | | **1** | | | 2 | | 0 | | | | | | | |
| 6 | | 1 | 1 | | | | 0 | **1** | | 0 | 2 | 0 | 0 | | 0 | | | | 0 | 0 |
| 7 | | 1 | 1 | 0 | | | | | **3** | | | | | | | | | | | |

| J/E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **8** | 0 | 2 | | | 2 | | | | 1 | | |
| 1 | 8 | **1** | | | | 2 | 1 | | | 1 | | |
| 2 | | | | | | 0 | | 0 | 1 | 1 | | |
| 3 | | 3 | | | | **7** | 1 | 2 | 2 | 1 | 3 | |
| 4 | 7 | 1 | 0 | **8** | **4** | | | 1 | 0 | 1 | | |

| J/E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **1** | 4 | 4 | | | | | 1 | 2 | 3 |
| 1 | 4 | **8** | 12 | 1 | 3 | 1 | 1 | 3 | 2 | 3 |
| 2 | | 1 | | **6** | | | | 1 | | |
| 3 | | 1 | | | | **6** | | | | 1 |
| 4 | | 3 | 3 | | 1 | | | | | |
| 5 | 4 | 2 | 1 | | | 3 | | **0** | 0 | 2 |

以上の表から次のようなことが考えられる。

・対訳関係にある記事対では、対応した文対の周辺に高い評価値の対応が現れる。
・対訳関係にある記事対では、表の対角線に近い位置に高い評価値の対応が現れる。

#### 5.4.2. 対訳関係にある記事対とない記事対の比較

文対応情報を導入する前のシステムでは、対訳関係にない記事対が不当に高い評価値を得てしまっている場合があった。この状況を分析するために、上記の文対応パターンの実験を、評価値の逆転が起こっている事例に適用してみた。

**対訳関係にない記事対**（第１候補：本来は第１候補になるべきではない）

| J/E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | 2 |
| 1 | | | | | | | | |
| 2 | | | | | | | 1 | 1 |
| 3 | 4 | 3 | 1 | 3 | 3 | | 3 | 2 |
| 4 | | 1 | 1 | 1 | 5 | 2 | 1 | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |

**対訳関係にある記事対**（第２候補：本来は第１候補になるべき）

| J/E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 2 | 2 | 1 | 2 | | 0 | | 1 | 0 | 1 | 2 | |
| 1 | 3 | **2** | 2 | 4 | 2 | | 1 | | 2 | 2 | 0 | | 1 |
| 2 | 3 | 2 | 1 | 2 | 1 | | 1 | | 2 | 2 | 1 | 2 | 2 |
| 3 | 2 | 4 | 3 | 3 | 2 | 0 | **4** | | 4 | 1 | 3 | 2 | |
| 4 | 2 | 0 | 3 | | 3 | | 1 | **4** | 0 | 4 | 3 | 3 | 1 |
| 5 | 3 | 1 | 1 | 0 | 0 | | 0 | **1** | 0 | 2 | 1 | 2 | 2 |
| 6 | | | | | | | | | | | | | |

**対訳関係にない記事対**（第６候補：本来も実際も評価値が低い）

| J/E | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | 0 | 0 | | | | | 1 | 1 | | | | |
| 1 | | 0 | 0 | | | | | | | | | 1 | |
| 2 | | 0 | | | | 0 | 0 | | 1 | | | 0 | |
| 3 | | 0 | 1 | | | 1 | 0 | 3 | 1 | | | 2 | |
| 4 | | 1 | | 0 | | 1 | 2 | 1 | 1 | 0 | 2 | 2 | 1 |
| 5 | | | | | | 0 | 0 | | 1 | 0 | 0 | | |
| 6 | | | | | | | | | | | | | |

この比較結果より、対訳関係にある記事対のみが、表の対角線周辺に高い評価値を持った文対を表していることがわかる。対訳関係にない記事対では、対角線とは無関係に高評価値が分布している。

このことから、記事全体の評価値を計算する際には、対角線周辺の記事対の評価が高くなるような重み付けをすることによって、評価値の信頼度を上げることができることがわかった。また、対角線周辺でも、特に記事の先頭に近いほど評価が高くなるような重み付けが必要となる。

システムでは､記事にあわせてこのようなチューニングを行い、組み込むことができる枠組みを作成した。

### 5.5. 対訳語句のフィードバック処理部

作成した対訳記事候補の中で、文対応の評価値の高い文対をアラインメントされた文対としてみなし、この文対の集合に対して、以前作成した対訳語句抽出システムを適用する枠組みを用意した。

今回対象とする対訳記事は、日英で記事長が大きく異なる場合があるため、文の先頭から順にアラインメントを行なうことが難しいと考えられる。そこで、一定の閾値以上の文対応度を持つ文対は対訳関係にあるとみなし、このような文対を収集した。

#### 5.5.1. 予備調査

１週間分の対象記事に対して作成した記事対応データから対訳語句を抽出した。これらの各記事対応候補の中で、ある文対応度以上の文対を対訳文対として抽出した。この文対の集合に対して、対訳語句抽出システムを実行した。

なお、対訳文から対訳語句を抽出するシステムは、当研究機関で開発済みであり、このシステムを利用して抽出を行った。また同様にして、対訳カタカナ語を抽出するシステムを利用して対訳カタカナ語一覧を作成した。

以下にその例をあげるが、これらの対訳語句を対応付けにフィードバックすることによって、対応付け精度が向上することが確認できている。

### 5.5.2. 対訳語句（複合語）候補の例

以下が上記のような条件で抽出した対訳語句の例である。共起頻度順にソートした時の上位15件である。

**対訳語句（複合語）候補の例**

| 英表記 | 英頻度 | 共起頻度 | 日頻度 | 日表記 |
|---|---|---|---|---|
| development of oil resources | 2 | 1 | 2 | 油田開発 |
| finance companies | 1 | 1 | 2 | ファイナンス会社 |
| residential zones | 4 | 2 | 4 | 住宅地域 |
| gulf war | 4 | 2 | 2 | 戦争終結後 |
| residential buildings | 4 | 2 | 6 | 住宅建設 |
| housing availability | 4 | 2 | 6 | 住宅環境 |
| interim report | 3 | 2 | 4 | 中間意見 |
| international contributions | 2 | 1 | 1 | 国際貢献策 |
| international relations | 6 | 2 | 4 | 国際関係 |
| chemical fertilizer plant | 4 | 2 | 4 | 化学肥料工場 |
| letter of credit | 6 | 2 | 6 | 債権回収 |
| market economy | 2 | 1 | 2 | 市場経済 |
| minesweeper issue | 3 | 1 | 1 | 掃海艇問題 |
| new technology | 3 | 1 | 1 | 新駅ビル |
| residential zone categories | 2 | 2 | 4 | 住宅地域 |
| revised bill | 5 | 2 | 6 | 見直し法案 |

**対訳語句（カタカナ語）候補の例**

| 英表記 | 英頻度 | 共起頻度 | 日頻度 | 日表記 |
|---|---|---|---|---|
| Bessmertnykh | 2 | 2 | 2 | ベススメルトヌイフ |
| Camp David | 2 | 2 | 2 | キャンプ・デービッド |
| Ganges | 1 | 1 | 1 | ガンジス |
| Irrawady | 1 | 1 | 1 | イラワジ |
| Lee Kuan Yew | 1 | 1 | 1 | リー・クアンユー |
| London summit | 1 | 1 | 1 | ロンドン・サミット |
| Marissa Blancada | 2 | 2 | 2 | マリサ・ブランカダ |
| Tu-Du | 2 | 2 | 2 | ツーズー |
| Quayle | 1 | 1 | 1 | クエール |
| Vegas | 4 | 4 | 4 | ベガス |
| care | 1 | 1 | 1 | クエール |
| dacoit | 2 | 2 | 2 | ダコイト |
| delta zone | 2 | 2 | 2 | デルタゾーン |
| feasibility study | 5 | 5 | 5 | フィージビリティ・スタディ |
| full | 1 | 1 | 1 | フル |
| laser radar | 2 | 2 | 2 | レーザーレーダー |
| power balance | 2 | 2 | 2 | パワー・バランス |
| shy | 1 | 1 | 1 | シャイ |
| smooth | 1 | 1 | 1 | スムーズ |

## 5.6. 対訳コーパス生成部

半対訳記事中に存在する対訳文候補は以下のパラメータの値が高い、という仮定のもとにチューニングを行ない、これをもとに対訳文を抽出するモジュールを作成した。。

- 英語形態素数と日本語形態素数で正規化した辞書対応度（辞書対応度も全体頻度で正規化）
- 英語の形態素数と日本語の形態素数の一致度

・ 文書中の対応位置による対応度
・文書同士の対応度

これは、分類としては「部分対応」の対訳記事で有効な手法である。「完全対応」の記事に関しては、ダイナミックプログラミングを利用して対応付けをしたほうが効率良く対応付けができる。これを適宜使い分けて対訳コーパスを作成した。

部分対応の記事に対する抽出を行い、順位付けした記事対を対応度の高い順に並べた。以下は、上記１週間の記事からの自動抽出結果である。

**生成された対訳コーパス**

| |
|---|
| 全国農業協同組合連合会（全農、鹿垣籾義会長）は九日、ヨルダン政府と合弁で同国アカバ市に大規模な化学肥料工場を建設する計画を明らかにした。<br>A major Japanese agricultural federation and Jordan will jointly build a large-scale chemical fertilizer plant in the Jordanian city of Aqaba, sources told the Mainichi Thursday. |
| ゴミ回収有料化に主婦の半数以上が賛成――経企庁調査<br>More than 50 percent of a surveyed group of housewives see charging households for waste disposal as inevitable, according to the Economic Planning Agency. |
| 会期延長なしの閉幕は、第百十二国会（八七年末から八八年五月）以来三年ぶり。<br>The ordinary session of the Diet which just ended on Wednesday broke the record for time spent on deliberations and adopting bills submitted by the Cabinet, according to a Liberal-Democratic Party Leader. |
| 「こんなにあっけなく死んでしまうなんて」と、結合体双生児のビン、タン姉妹を取材した社会部のO記者。<br>Conjoined Viet Twins Die Before Operation |
| 最近では三菱石油がベトナム沖の油田開発に参加を表明したし、日本航空もベトナム線開設へ動き出している。<br>Mitsubishi Oil Co. is about to participate in the development of oil resources off the Vietnam coast, and the opening of a Vietnam route is being planned by Japan Airlines. |
| 中国本土からの亡命者へ優待規定停止――台湾国防部<br>Editorial--Taiwan And The Mainland |
| 交流の芽に法の壁、フィリピン女性の留学申請却下、偽装留学防止の法改正で法務省<br>The Justice Ministry has applied an ordinance that restricts access to the nation's schools to official foreign exchange and Japanese-language students to prevent a Filipino high schooler from studying here. |
| 祖母の放火で、２人の孫が焼死　おかずで息子と口論、自宅に灯油――栃木・日光<br>NIKKO, Tochigi -- Two young children were burned to death over the weekend after their grandmother allegedly set fire to the house in a quarrel with her son, it was reported Sunday. |
| 経済人の間でベトナムが脚光を浴びている。<br>Vietnam has suddenly begun to draw attention. |
| 「こんなにあっけなく死んでしまうなんて」と、結合体双生児のビン、タン姉妹を取材した社会部のO記者。<br>The sisters, 6-month-old Binh and Thanh, were believed to have been born conjoined due to their parents' exposure to the remnants of a defoliant used in the Vietnam War. |

## 6. システム出力

### 6.1. 記事対応付け

本研究で作成したシステムは、利用する対訳情報や、チューニングの進み具合で精度が大幅に変わるため、ここでは特に定性的な結果を中心に説明する。

以下は､記事対応付けの結果である。予備調査の項で行った１週間分の人手見積りと同じ記事を使っている。ここで、「対応」とは、人手で見つけた対応であり、前述の予備調査と同じ印で示している。これに対して、「正誤」では、この人手正解に照らし合わせて、正解かどうかを判定している。

●：本来の正解日本語記事を自動抽出で取り出した
▲：本来の正解日本語記事を自動抽出で取り出したが、閾値には達しなかった
×：本来の正解日本語記事を自動抽出できなかった
○：本来抽出されるべきでない日本語記事を自動抽出で取り出さなかった。
△：本来抽出されるべきでない日本語記事を自動抽出したが、閾値を変えれば振り落とせる。
×：本来抽出されるべきでない日本語記事を自動抽出した。

この結果から、対応がもともとしっかりしている記事対はほとんど誤りなく抽出できていることがわかる。また、対応度が低いものに対しては、閾値を調節することによって十分取り出せることがわかる。

**システムの記事対応付け結果**

| 日付 | 記事の題名 | 対応 | 正誤 |
|---|---|---|---|
| 1991/05/05 | Holiday Flea Markets Mushroom | | ○ |
| 1991/05/05 | Environment Concern Rises With Age | | ○ |
| 1991/05/05 | Long-Term Rail Plan Formulated | | △ |
| 1991/05/05 | Seven Wonders Of Japan--7 Types Of Japanese Ambiguity | | ○ |
| 1991/05/05 | School Days (14): Roles | | ○ |
| 1991/05/05 | Fashion--A Look At The Tokyo Collections | | ○ |
| 1991/05/05<br>1991/05/05 | Child Population Hits Lowest Recorded Level<br>子供の数減る、２２１５万３０００人、人口の１７．９%に | ◎ | ● |
| 1991/05/05 | Foreign Workers Promised Better Job Conditions | | ○ |
| 1991/05/05<br>1991/05/04 | Housewives Foresee Paying For Waste Disposal<br>ゴミ回収有料化に主婦の半数以上が賛成－－経企庁調査 | ○ | ● |
| 1991/05/05<br>1991/05/04 | Sudden Illness Claims Workers in Their Prime<br>働き盛りの死、８人に１人は突然死　男性は女性の３倍－－厚生省初調査 | ○ | ● |
| 1991/05/05<br>1991/05/05 | Waseda Hostages Come Home<br>パキスタンの誘惑事件の早大生３人が帰国　「軽率」「反省」「無謀」と語る | △ | ▲ |
| 1991/05/05<br>1991/05/04 | Editorial--Taiwan And The Mainland<br>中国本土からの亡命者へ優待規定停止－－台湾国防部 | △ | ● |
| 1991/05/06<br>1991/05/04 | Editorial--Vietnam Trade In Limelight<br>［社説］対越経済協力、半歩踏み出せ | ◎ | ● |
| 1991/05/06 | Film--Last Frankenstein | | ○ |
| 1991/05/06 | Bunraku--May Preview | | ○ |
| 1991/05/06<br>1991/05/05 | 2 Children Die In Fire Set After Family Tiff<br>祖母の放火で、２人の孫が焼死　おかずで息子と口論、自宅に灯油－－栃木・日光 | △ | ▲ |
| 1991/05/06<br>1991/05/05 | Kids' Pocket Money Increase 13 Percent<br>１年間のお小遣い、２年前の１３%増加　貯蓄も増えた－－日本生命調査 | ○ | ● |
| 1991/05/06<br>1991/05/05 | Law Restricts Schools To 'Official' Foreign Students<br>交流の芽に法の壁、フィリピン女性の留学申請却下、偽装留学防止の法改正で法務省 | △ | △ |
| 1991/05/06<br>1991/05/05 | Ministry Looking Into Housing Zone Changes<br>住宅地に「中高層専用区」、建設推進へ高さの制限を撤廃－－建設省方針 | △ | ● |
| 1991/05/06 | Things To Do--Kansai | | ○ |
| 1991/05/06<br>1991/05/06 | Tokyo Firm Defaults On 4.5 Mil. Dollars Owed To Soviet<br>ロケット発射に支障　「早く機材返して」とソ連側－－宇宙商法倒産トラブル | △ | ● |
| 1991/05/06 | Things To Do--Kanto | | ○ |

| | | | |
|---|---|---|---|
| 1991/05/06 | Japan How To--Woodblock Printmaking (5) | | ○ |
| 1991/05/06 | Mingei--Itaya-Zaiku | | ○ |
| 1991/05/06 | The Metropolitan Library Service In Tokyo. | | ○ |
| 1991/05/06 | Tokyo Univ. Hospital's Professionalism Under Fire: Series I | | ○ |
| 1991/05/08 | Within My Ken--The Decayama, Toyama | | ○ |
| 1991/05/08 | Vanishing--Geisha | | ○ |
| 1991/05/08 | Tokyo Univ. Hospital (2): A Mecca For Unofficial Doctors | | ○ |
| 1991/05/08<br>1991/05/06 | B'desh Embassy Calls For Aid<br>政府発表の犠牲者も１２万５千７百人に――バングラデシュのサイクロン | △ | ▲ |
| 1991/05/08 | Along The Tokaido--Day Three | | ○ |
| 1991/05/08<br>1991/05/08 | Coke Carrier Arrested<br>ボリビアからコカイン５・７キロ　成田空港で台湾人逮捕――過去最高の押収量 | △ | ● |
| 1991/05/08 | House Member Commits Suicide | | ○ |
| 1991/05/08<br>1991/05/06 | End To Labelling Chaos Sought: Guidelines For Vegetables<br>「有機」や「無農薬」などの乱立表示にガイドライン設置へ――農水省方針 | ○ | ▲ |

## 6.2. 文対応付け（対訳コーパス作成）

記事対応付けによって、「完全対応」となった記事対に対して、文対応付けを行った。ほとんどのものについて、実用的な精度が得られた。以下の例では、記事の途中から若干のずれが生じているが、対訳語句抽出や対訳フレーズ獲得などの実用面では十分な性能が得られている。

**システムの作成した対訳コーパス**

| 英語文 | 日本語文 | 対応 |
|---|---|---|
| Editorial--Vietnam Trade In Limelight | ［社説］対越経済協力、半歩踏み出せ | ○ |
| 1991/05/06 | 1991/05/04 | ○ |
| Vietnam has suddenly begun to draw attention. | 経済人の間でベトナムが脚光を浴びている。 | ○ |
| Planes flying from Bangkok to Ho Chi Minh City are reported to be filled with businessmen. | バンコク発ホーチミン市行きの航空便はビジネスマンたちでいっぱいだそうだ。 | ○ |
| Mitsubishi Oil Co. is about to participate in the development of oil resources off the Vietnam coast, and the opening of a Vietnam route is being planned by Japan Airlines. | 最近では三菱石油がベトナム沖の油田開発に参加を表明したし、日本航空もベトナム線開設へ動き出している。 | ○ |
| Discussion are taking place within the Asian Development Bank for the restart of financing Vietnam which has begun to convert from a planned to a market economy. | アジア開発銀行の内部でも、計画経済から市場経済の方向へ転換を始めたベトナムに対し、融資を再開しようとする議論が出ている。 | ○ |
| We believe that the time has come to begin thinking seriously, in a forward looking manner, about economic cooperation with Vietnam. | 私たちは対越経済協力を慎重かつ前向きに見直す時期が来たと考える。 | ○ |
| In the background of the new world attention to Vietnam are that country's adoption of a policy of economic reforms, activation of a market economy, and advance along the path of competition by the introduction of a system of ownership inclusive of private ownership. | ベトナムが世界から注目され始めた背景には、過去数年ドイモイという経済刷新政策をとり、市場メカニズムの活用、私有制を含む所有形態の導入による競争を推進していることがある。 | ○ |
| The results of these measures have not been fully substantiated as yet, but in 1989 the export of 1,1400,000 tons of rice become possible. In fact, Vietnam become the world's third largest rice exporting country next to the United States and Thailand. | その成果はまだ十分あがっているとはいえないが、八九年には一四〇万トンのコメの輸出が可能となり、米国、タイにつぐ世界第三位の輸出国となった。<br>この国が経済躍進の目覚ましいタイと並んで広義の「インドシナ経済圏」形成の可能性を持っていることも注目に値する。 | △ |
| It is worthy of note that Vietnam has the possibility of becoming the core of an "Indochina economic bloc" that will be on a par with Thailand which has made a remarkable economic advance. | ベトナム、ラオス、カンボジアの旧仏領インドシナ連邦にタイ、ミャンマー（旧ビルマ）を加えた五カ国は人口約一億七〇〇〇万人。 | ○ |

| | | |
|---|---|---|
| Five countries, including Indochina's Vietnam, Laos and Cambodia, together with Thailand and Myanmar(formally Burma) is five times the size of Japan. | その面積は日本の五倍で、メコン、イラワジ両川の流域は豊かな水資源、土壌（デルタゾーン）のほか、開発可能な鉱物資源に恵まれている。 | △ |
| The basins of the Mekong and Irrawady rivers haveabundant water resources and a delta zone, and there are mineral resources that could be developed. | ベトナムが市場経済の方向へ歩み出し世界市場へ向け窓を開こうとしているのは、東西冷戦の終焉とソ連・東欧からの援助の縮小という現実とも関係があろう。 | × |
| Vietnam's move toward a market economy is related to the end of the Cold War and the reduction of aidfrom the Soviet Union and Eastern Europe. | この国と東南アジア諸国連合（ＡＳＥＡＮ）との関係強化が急ピッチで進みつつあることにも目を向けるべきだ。 | △ |
| Meanwhile, Vietnam's relations with the countries of ASEAN are being strengthened at a rapid tempo. President Suharto of Indonesia visited d Vietnam in November last year. | 昨年十一月にはスハルト・インドネシア大統領が訪越し、この時、ベトナムはＡＳＥＡＮへの加盟を正式に表明した。 | △ |
| As that time Vietnam expressed its desire to join ASEAN. Although some time might be required before this can take place, the feeling is spreading among the ASEAN countries that "it is not desirable To isolate Vietnam." | 加盟に至るまでまだ長い時間が必要だろうが、ＡＳＥＡＮ諸国内部には「ベトナムの孤立化は好ましくない」との判断が広がりつつある。 | △ |

## 7. 結論

日本語および英語の新聞記事からなるコンパラブルコーパスから、対訳文対で構成される対訳コーパスを作成する手法を開発し、その有用性を検証した。対象とするコンパラブルコーパスとしては、人手による翻訳が施された対訳文も一部に混在していることを前提としているが、そのようなコーパスの代表例として新聞記事があげられる。本手法においてはまず一方の言語で書かれた文書を一つずつ取り出し、それぞれについて、対訳関係にあると思われるもう一方の言語で書かれた文書を抽出した。ここで、対訳関係にあるというのは、完全な対訳文書である場合もあるが、一般的には内容的に関連が深く、対訳語句が多く含まれた文書対を意味する。このような関連文書には、部分的に対訳文、あるいはほぼ内容の同じ文が含まれることが多く、そのような対訳文を関連文書の中から抽出し、再度関連度を確認したうえで、対訳文として登録した。

今回用いたコンパラブルコーパス中の関連度の高い日本語記事と英語記事の間には大別して３通りの関係が認められた。

１． 完全対応：記事中の全ての文が過不足なく対応しているような文書対。言語Ａによって記述される文書中の全ての文が、言語Ｂによって記述される文書中に対訳関係にある文を持っている。
２． 部分対応：文書対において、一部の文が対訳関係となっている。対訳文に関しては、その部分を抽出すれば対訳コーパスとして使える。
３． 内容対応：対訳文は存在しないが、内容的には同等な文書対。対訳コーパスとして使える部分はないが、対訳語句や対訳表現は抽出することができる。

対象記事は、もともとが日本語で書かれたもので、その一部を後で英語に翻訳しているため日本語記事のほうが英語記事よりも本数がかなり多い。このため、日英関連記事対を求めるのには英語記事から対応する日本語記事を探した方がその逆よりも効率が良い。したがって英語からの検索システムを作成した。記事関連付けのシステムは、英語キーワード抽出部、日本語キーワード抽出部、キーワード言語変換部、記事対応付け部、および文対応付け部から構成されている。基本構成における対訳コーパス作成システムは、一般的なコンパラブルコーパスに対して自動処理できる処理を実現したものであるが、実際の対訳コーパス作成では、ある特定の種類のコンパラブルコーパスに対して継続的に処理を行なうことが多く、その対象に合わせて人手でチューニングを行なう必要がある。たとえば、システムが自動的に作成した対訳情報の候補から、正しいと思われるものを選択して対訳情報データベースに蓄積することによって、対訳情報が充実してくる。対訳情報がある程度充実すれば、その後はシステムは自動に近い形で運用することができるようになる。このように、最終的に出来上がったシステムは完全自動で動くように設計されているが、システムの立ち上げ時や、高精度の結果が欲しい場合には、必要に応じて人手でチューニングできるようになっている。　チューニング対象となる対訳情報としては、対訳語句、対訳カタカナ語用フィルタ、発信地リストなどがある。逆に、蓄積された対訳情報は、形態素解析、キーワード変換、記事対応付け、文対応付けなどの幅広いモジュールで使われる。対訳情報の充実がこれらモジュールの精度向上につながる。

次に、記事対応付けによって「完全対応」となった記事対に対して、文対応付けを行った。文の対応付けは基本的に記事対応付けとほぼ同じ手順で行う。今回は、使用したコーパスの特徴に即したヒューリスティックスを導入することにより、大幅に記事対応および文対応の精度を上げられることがわかった。すなわち、記事全体の評価値を計算する際に、文対応マトリックス上の対角線周辺の記事対の評価値が高くなるような重み付けをすることによって、評価値の信頼度を上げることができることがわかった。また、対角線周辺でも、特に記事の先頭に近いほど評価が高くなるような重み付けすることが有効であることが確認された。文対応付けの精度を厳密に評価するにはそれ専用の評価用データが必要なため、今回の検討の範囲外であるが、人手による簡単な評価を行った結果、文対応付けに関しては、対訳語句抽出や対訳フレーズ獲得などの実用面では十分な性能が得られており、本手法が十分有効であることが確認できた。

## 成果発表、特許等の状況

## 購入機器一覧

## 付録

### 参考文献


Koji Tsukamoto, Manabu Sassano, Kunio Matsui, “Taggers for Unknown Words using Decision Tree and Lazy Learning”, Proceedings of 5th NLPRS, 1999

塚本浩司、「富士通研究所IREX NE参加システム」、Proceedings of the IREX Workshop, 1999

塚本浩司、「有限状態変換器の誤り駆動型学習を用いた固有表現抽出」、情報処理学会自然言語処理研究会、1999

大倉清司、富士秀、潮田明、「情報検索装置および方法」、特願平10-69050,1999

潮田明、富士秀、「情報検索装置及び情報検索方法」、特願平10-21631,1998

Akira Ushioda, “Hierarchical Clustering of Words and Application to NLP Tasks”, Proceedings of the Fourth Workshop on Very Large Corpora, p28-41, 1996

Akira Ushioda, “Hierarchical Clustering of Words”, Proceedings of the 16th International Conference on Computational Linguistics, p1159-1162, 1996

潮田明、「単語・連語分類処理方法、連語抽出方法、単語・連語分類処理装置、音声認識装置、機械翻訳装置、連語抽出装置及び単語・連語記憶媒体」、特願平9-167243, 1996

高尾哲康、富士秀、松井くにお、「対訳テキストコーパスからの対訳情報の自動抽出」、情報処理学会自然言語処理研究会、p51-58, 1995

大倉清司、「タグ付きコーパスからの統語・意味的知識の自動獲得」、電子情報通信学会ＮＬＣ技術研究報告、p9-14, 1995

電子・情報分野
99Y補 03-110-5

平成11年度

新エネルギー・産業技術総合開発機構

提案公募事業（産学連携研究開発事業）

研究成果報告書

# 複数言語にまたがる言語知識処理技術の研究

## （オントロジーを利用した検索技術の研究）

平成13年3月

**（株）東芝**

平成１１年度 新エネルギー・産業技術総合開発機構 提案公募研究開発事業
産学連携研究開発事業） 研究成果報告書概要

| | |
|---|---|
| 作成年月日 | 平成 13年 3月 31日 |
| 分野／プロジェクト ID 番号 | 分野：電子・情報分野<br>番号：99Y 補 03-110-5 |
| 研 究 機 関 名 | (株)東芝 |
| 研究代表者部署・役職 | 研究開発ｾﾝﾀｰ ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 室長 |
| 研究代表者名 | 平川秀樹 |
| プロジェクト名 | 複数言語にまたがる言語知識処理技術の研究<br>（オントロジーを利用した検索技術の研究） |
| 研究期間 | 平成 12年 3月 15日～平成 13年 3月 29日 |
| 研究の目的 | 現在実用化されている検索技術では、利用者が指定したキーワードもしくはその同義語を利用して検索を行っているため、概念的には類似性があったとしても、言葉として検索キーワードと類似しているものを含む文書でなければ検索できないという問題がある。そこで本研究では、概念レベルでの類似性を評価して検索を行う新たな検索技術を開発することを目標とする。 |
| 成果の要旨 | オントロジーを用いた概念検索システム、オントロジー生成システムをそれぞれ開発し、これを用いて特許文書、ならびに日本語検索テストデータセット（BMIR－J2）において、検索語彙拡張の実験を実施した。<br>その結果、BMIR－J2において、汎用オントロジー技術により再現率が約１１%向上することを確認した。またオントロジー生成技術により再現率が約３%向上することを確認した。（いずれも、語彙拡張前と比べ、正解文書の順位の平均値も低下していないことを確認済み。） |
| キーワード | オントロジー、シソーラス、検索、概念、キーワード、拡張 |
| 成果発表・特許等の状況 | 成果発表 1件 |
| 今後の予定 | オントロジー知識をより詳細化し、意味関係を明確化するための技術の開発を検討する。<br>また複雑な検索課題での効果を高める方式の開発のため、言い換え(paraphrase)の研究を進め知見を蓄積する。 |

Summary of R&D Report for FY 1999 Proposal-Based R&D Program
of New Energy and Industrial Technology Development Organization

| | |
|---|---|
| Date of preparation | March 31, 2001 |
| Field / Project number | Field: Electronics and information technology<br>No.99Y03-110-5 |
| Research organization | TOSHIBA Corporation |
| Post of the research coordinator | Laboratory Leader of Human Interface Laboratory,<br>Corporate Research & Development Center |
| Name of the research coordinator | Hideki Hirakawa |
| Title of the project | Multilingual natural language processing technologies<br>(Research of information retrieval using ontology) |
| Duration of the project | March 15, 2000 ～ March 29, 2001 |
| Purpose of the project | The practical information retrieval systems only use keywords and synonyms of keywords. For this reason, if a document, which is conceptually similar to a query, does not contain any synonyms of keywords, it cannot be searched. This project aims at developing a new information retrieval system while evaluating the conceptual similarity. |
| Summary of the project | We developed the information retrieval system using ontology, and the ontology generating system using a corpus. We evaluated these systems for patent documents and the Japanese test-data set (BMIR-J2).<br>As a conclusion, in BMIR-J2, the recall ratio improved about 11% with general ontology. The recall ratio improved about 3% with generated ontology. While the average of the ranking of the correct documents is not falling. |
| Key Word | ontology, thesaurus, concept, information retrieval, query, expansion |
| Publication, patents, etc. | 1 technical report. |
| Future plans | It is necessary to develop the technique for classifying the semantic relation between words. |

# まえがき

本研究では、クロスランゲージ検索のベースとなる検索技術そのものを高精度化するための研究を行う。
現在実用化されている検索技術では、利用者が指定したキーワードもしくはその同義語を利用して検索を行っているだけである。そのため、概念的には類似性があったとしても、言葉として検索キーワードと類似しているものを含む文書でなければ検索できないという問題がある。そこで本研究では、概念レベルでの類似性を評価して検索を行う新たな検索技術を開発することを目標とする。
本研究では、世界知識をオントロジーとして表現し、それを利用した検索技術の研究を行う。検索でオントロジーを利用するには、オントロジーを適切に表現するための枠組みが必要であると同時に、大規模なオントロジーを構築するための技術が必要となる。したがって、オントロジーを生成するための関連技術の研究を合わせて行う。

## I.　オントロジー技術による概念検索の実証

これまでオントロジー技術を利用した概念検索の効果はあまり報告されておらず、実証研究の意義は大きなものである。
研究を行うためには大規模なオントロジーが必要となる。日本語における大規模な語彙ＤＢとしてはEDR 辞書[EDR 1995]が知られている。
ＥＤＲ辞書は人の判断に基づいて作成された辞書であり、ここで提供される日本語単語辞書と概念辞書を用いることにより、体系化された日本語オントロジーを構築することができる。しかし、ＥＤＲ辞書は機械翻訳システム等での選択制限の問題解決には一定の効果を発揮しているが、概念検索での効果は検証されておらず、今回は概念検索を目的とした利用方法を開発し、その上で効果の検証を行った。
EDR辞書を改良することにより大規模なオントロジーを構築する手法を通して、オントロジー技術の汎用化を図る。

## II.　コーパスを利用したオントロジー生成技術

オントロジー技術は、これまで主に知識を表現する枠組み（すなわち、専門分野の知識が形式的に記述できるか）の観点から研究がなされてきたため、知識の記述そのものは人手で行っている。そのため、現状のままでは大規模な分野に適用するには、オントロジーの作成にコストがかかりすぎ、実用的に利用することは困難である。そこで本研究では、オントロジーを効率よく低コストで構築できるよう、専門分野のコーパスから知識を抽出するためのオントロジー生成技術を開発する。

# 研究者名簿


研究代表者　平川秀樹　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 室長
連絡先 〒212-8582 神奈川県川崎市幸区小向東芝町１
電話: 044-549-2020，FAX: 044-520-1308

研究者　住田一男　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 主任研究員

木村和広　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 研究主務

大嶽能久　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 研究主務

木下　聡　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 研究主務

小野顕司　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 研究主務

齋藤佳美　株式会社東芝 研究開発センター ﾋｭｰﾏﾝｲﾝﾀｰﾌｪｰｽﾗﾎﾞﾗﾄﾘｰ 研究主務

# 目次

# 1. オントロジー構築と検索語彙拡張実験の方式

## 1.1. 概要

Mandala らは、英語の検索課題（TREC－7）に対し、WordNet、およびコーパスから自動獲得した類義語ペアを用いて検索語彙を拡張する研究を行っており、検索性能の向上が報告されている[Mandala 2000]。
本研究での実験は、この方式を日本語に対して適用したものである。具体的には、検索課題として日本語のテスト・コレクションであるBMIR－Ｊ２を用い、汎用オントロジーとしてはＥＤＲ辞書を用いた。また、これとは別に検索課題として特許の引例検索をタスクとする実験も行っている。
以下に、検索方式、ＥＤＲ辞書の利用方法、コーパスからのオントロジーの自動獲得方法について説明する。

## 1.2. 検索語彙拡張方式

ベクトル空間法によりクエリー $q$ はベクトル $\vec{q} = (w_1, w_2, ..., w_n)$ で表される。ただし各 $w_i$ はクエリー $q$ に含まれる各探索語 $t_i$ の重みである。初期のクエリーの重みは下記の式によって得られる。

$$\frac{(\log(tf_{ik}) + 1.0) * \log(N / n_k)}{\sqrt{\sum_{j=1}^{n} \left[ \left( \log(tf_{ij}) + 1.0 \right) * \log\left( N / n_j \right) \right]^2}}$$

ただし $tf_{ik}$ はクエリー $q_i$ 中の語 $t_k$ の出現頻度であり、 $N$ は文書集合中の総文書数であり、$n_k$ は語 $t_k$ が含まれる文書の数である。
本手法では、複数辞書を組み合わせて類似度を判定することにより、各種の辞書を相互に補完する。その際、各種の（オントロジー）辞書における類似度には必ずしも固定した値の範囲が無いため、類似度を[0,1]の範囲に収める手法として、各タイプのオントロジーに下記の正規化戦略を適用する。

$$sim_{new} = \frac{sim_{old} - sim_{\min}}{sim_{\max} - sim_{\min}}$$

クエリー $q$ と語 $t_j$ との間の類似度は下記のように定義される。

$$simqt(q, t_j) = \sum_{t_i \in q} w_i * sim(t_i, t_j)$$

ただし $sim(t_i, t_j)$ の値は上記の結合されたオントロジーから得られるものである。
これにより、クエリー $q$ に関して文書集合中の全ての語を、それらの $simqt$ に従ってランク付けすることができる。拡張語 $t_j$ は $simqt(q, t_j)$ の値の高いものである。
拡張語 $t_j$ の重み $weight(q, t_j)$ は $simqt(q, t_j)$ の関数として下記のように定義される。

$$weight(q, t_j) = \frac{simqt(q, t_j)}{\sum_{t_i \in q} w_i} \quad \text{ただし } 0 \le weight(q, t_j) \le 1.$$

拡張語の重みはクエリーに現れる全ての語と語の間の類似度の両方に依存し、値の範囲は0と1の

間である。
クエリー $q$ は下記のクエリーを追加することによって拡張される。

$$\vec{q}_c = (a_1, a_2, ..., a_r)$$

ただし$a_j$ は $t_j$ が上位 $r$ 番以内にランクされている語に属しているならば $weight(q, t_j)$ に等しく、それ以外は 0 である。
結果として得られる拡張クエリーは、

$$\vec{q}_{\exp anded} = \vec{q} \circ \vec{q}_e$$

となる。ただし $\circ$ は結合演算子として定義される。

## 1.3. EDR辞書の利用

### 1.3.1. EDR辞書による単語間類似度定義

EDR 辞書[EDR 1995]は人の判断に基づいて作成された大規模語彙 DB であり、ここで提供される日本語単語辞書と概念辞書を用いることにより、WordNet ライクな体系化された日本語オントロジーを構築することができる。すなわち、synset(同義語集合)に対応するものとして、日本語単語辞書において同一の概念 ID を付与された表記集合を考え、体系を構成する synset-ID に対応するものとして概念 ID を考えればよい。
図 1において、*c1-cm*は概念体系を構成する概念ID であり、*w1-wn*は日本語単語辞書を構成する単語表記である。概念体系は、*c1-cm* 間に成立する *isa* 関係を列挙したものであり、日本語単語辞書は、*w1-wn* と*c1-cm* の対応関係を列挙したものである。

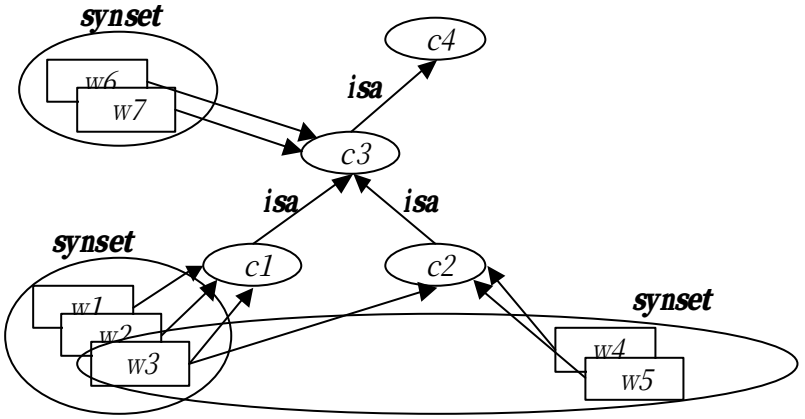



**図 1 日本語オントロジーとしての EDR 辞書**

このように EDR 辞書を日本語オントロジーとして捉えることにより、2 語の意味的類似度に対して以下に示す2つの尺度を考えることができる。すなわち、

1. パス長ベースの類似度
2. 情報量ベースの類似度

である。

**パス長ベースの類似度**

語 *w1* と語 *w2* のパス長ベースの類似度は、*w1* の各語義から *w2* の各語義へのパスのうち、最短のパス長と定義する(Leacock and Chodorow 1998)。すなわち、

$$sim_{path}(w1, w2) = \max[-\log(\frac{Np}{2D+1})]$$

ここで、*Np* は、*w1* から *w2* に至るパス中に存在する概念ノード数であり、*D* はオントロジーの最深ノードの深さである。EDR 辞書では、*D*=16 である。

**情報量ベースの類似度**

パス長ベースの類似度は、各パスが均一の意味的距離を保持しているという前提の上に成り立っている。しかし、この前提が現実的でないことは従来から指摘されており、Resnik は、この問題に対し、概念のもつ情報量という観点から類似度の再定義を行った[Resnik 1995]。すなわち、

$$sim_{IC}(w1, w2) = \max_{C \in S(c1,c2)} [-\log\ p(c)]$$

ここで、S(c1,c2)は概念 *c1* と *c2* を包含する概念集合である。概念出現確率 *p(c)*は、文書集合から得られる相対頻度から計算される。すなわち、

$$p(c) = \frac{freq(c)}{N}$$

ここで、*N*は文書集合で観察された全単語(但しEDRの概念クラスに属さないものを除く)の頻度である。また、*freq(c)*は、単語 *w* が1回観察されたとき、*w*のすべての多義 *Sense*(*w*)={*c1,…,cn*}とこれら多義の各上位概念集合 *Ancestors*(*ci*)との和集合 *SetOfAncestors*(*w*)の各要素が 1 回出現したものとしてカウントする。

$$SetOfAncestors(w) = (\bigcup_i Ancestors(c_i)) \cap Sense(w)$$

**類似度の和**

上記、パス長ベースの類似度と情報量ベースの類似度の和により、EDR 辞書ベースの類似度を定義する。

$$sim_{sum}(w1, w2) = sim_{path}(w1, w2) + sim_{IC}(w1, w2)$$

### 1.3.2. 類似度DBの作成

上記の各定義に基づく類似度は、検索実験の効率的実行の観点から、実験に先立ち、表形式の DB として事前に作成しておく。DB の作成にあたっては、

1. 概念体系のトリミング
2. DB 化

の 2 段階にて実施した。

**概念体系のトリミング**

EDR 概念体系は、約 40 万ノードの概念から構成されるが、このうち英語由来の概念など、特に情報量ベースの類似度計算において無用(かつ有害)な概念ノードが多数(20 万ノード以上)存在する。これらは、不要リーフノード集合を初期値として与えることによって、トリミングすることが可能である[Kimura and Hirakawa 2000]。今回の実験にあたっては、以下の 2 種の限定を加えるよう、不要リーフ

ノード集合を与えトリミングを実施した。

1. 現代語頻出名詞概念限定: EDR 日本語単辞書第 2 版からは、古語由来の概念に対し、それを示すコード"OLD "が付与されている。また、EDR 日本語コーパスにおける概念頻度が付与されている。これらの情報を用いて、日本語名詞由来かつ現代語(概念)かつ頻出概念(今回は頻度 1 以上とした)である概念に限定した、概念体系及び日本語単語辞書を作成した。
2. 日英翻訳辞書見出し限定: 検索文書集合のインデキシングには、日英翻訳システムで用いられている形態素解析部を利用している。このため、EDR 辞書の見出し語集合と翻訳辞書の見出し語集合との積集合に限定した、概念体系及び日本語単語辞書を作成した。ただし、EDR の見出し語集合並びに概念集合は、一般語のみならず専門用語辞書のエントリーもマージすることで作成した(前記第 1 のトリミングでは、一般語のみを対象とした)。なお、専門用語体系のトップノード(2 概念)は、一般語体系のルートノード直下に配置した。なお、本オントロジーでは、このフルスペック版に加え、名詞限定版、動詞限定版も作成した。

このようにして、トリミングされたEDR 辞書ベースオントロジーのプロパティを図 2に示す。

| | オントロジー名 | 単語数(表記異なり数) | 概念数 |
|---|---|---|---|
| (I) | 現代語頻出名詞概念限定 | 43,734 | 49,566 |
| (II) | 日英翻訳辞書見出し限定 | 72,248 | 102,197 |
| (III) | 名詞限定 | 60,231 | - (フルスペック版を流用) |
| (IV) | 動詞限定 | 18,142 | - (フルスペック版を流用) |

**図 2 EDR 辞書ベースオントロジーのプロパティ**

## DB 化

単語間類似度 DB は、具体的には、Berkley DB を用いたハッシュ表であり、図 3の構造をもつ。キーとしては、単語 1 語から成るキーと単語2 語から成るキーの 2 種類がある。単語1 語から成るキー(例えば *w1*)に対しては、このキーとの類似度の降順に整列された有限個の単語リスト(*w11,w12,w13,…*)が格納されている。また、単語2 語から成るキー(例えば *w1,w2*)に対しては、この 2 語間の類似度(*sim12*)が格納されている。

| key | value |
|---|---|
| *w1* | *w11,w12,w13,…* |
| … | … |
| *w1,w2* | *sim12* |
| … | … |

**図 3 単語類似度 DB の構成**

本 DB の作成にあたっては、予めすべての単語の組み合わせに対して計算しておいてもよいが、これには莫大な計算コストを要するため、今回は、各検索課題(特許,BMIR-J2,IREX)ごとに、それぞれ質問語集合 $Q$ と索引語集合 $I$ を予め与え、この直積 $Q \times I$ の単語ペアのみに対して DB を作成した。概念出現確率の計算も、各検索課題の検索対象文書集合を用いて行った。

## 1.4. オントロジーの自動獲得

### 1.4.1. 係り受け関係をベースとしたオントロジー知識の獲得

コーパス（大量文書）が与えたれた時に、そこからオントロジー知識を獲得する方法として、単語間の共起頻度を利用する方法がある。今回は係り受け関係での共起をベースとした方法によって辞書を自動生成している。
辞書の作成方法は、次の通りである。

① コーパス中の文を構文解析し、係り受け関係にある名詞―動詞の対を抽出する。
今回の実験では、名詞が動詞のｗｏ格になっている場合を抽出した。（ｗｏ格以外の係り受け関係についても、一部実験を行っている。）

② 抽出した名詞―動詞ペアに対し、次のような相互情報量を算出する。

$$I_r(v_i, n_j) = \log \frac{f_r(n_j, v_i) / N_r}{(f_r(n_j) / N_r)(f_r(v_i) / N_r)}$$

ただし、ここで$r$ は①で抽出した名詞－動詞対のかかり受け関係を指す。$f_r(n_j, v_i)$ は名詞$n_j$ が動詞 $v_i$ と関係 $r$ の係り受け関係で出現した頻度、$f_r(n_j)$ は名詞$n_j$ が任意の動詞と共に関係 $r$ の係り受け関係で出現した頻度、$f_r(v_i)$ は動詞$v_i$ が任意の名詞と共に関係$r$ の係り受け関係で出現した頻度のことであり、$N_r$ は関係$r$ の係り受け関係の出現頻度のことである。例えば、①でｗｏ格を抽出した場合、関係$r$ のかかり受け関係とはｗｏ格のかかり受け関係のことである。

③ 抽出した名詞間、および動詞間の類似度 sim1 は、次の式により定義する。

$$sim1(w_1, w_2) = \frac{\sum_{(r,w) \in T(w_1) \cap T(w_2)} (I_r(w_1, w) + I_r(w_2, w))}{\sum_{(r,w) \in T(w_1)} I_r(w_1, w) + \sum_{(r,w) \in T(w_2)} I_r(w_2, w)}$$

上の式で、$r$ は①で抽出した名詞－動詞対のかかり受け関係を指す。$w$ は$w_1$、$w_2$ のかかり受け先の単語（$w_1$ と$w_2$ が名詞の場合は動詞、$w_1$ と$w_2$ が動詞の場合は名詞）である。$T(w')$ は$w'$ が関係$r$ で係り受けする単語$w$ の集合（ただし$I_r(w, w') > 0$）
すなわち、分子は2つの名詞（もしくは動詞）に共通する共起動詞（もしくは名詞）との相互情報量の和を求めている。分母はそれぞれの名詞（もしくは動詞）の相互情報量の総和を求めている。

### 1.4.2. 類似度の定義の追加

上記の方式に対し、類似度の定義を次のように変更した類似度 sim2 を追加した。

$$sim2(w_1, w_2) = \sum_{(r,w) \in T(w_1) \cap T(w_2)} (I_r(w_1, w) + I_r(w_2, w))$$

### 1.4.3. 作成した辞書の諸元

上記の方法を用いて、新聞記事検索、および特許検索向けに、それぞれ、新聞記事、特許をコーパスとした数種類の自動獲得辞書を作成した。
作成した辞書は次の通りである。

| 辞書名 | コーパス | 類似度 | 関係 | sim条件 | 類義語ペア数 |
|---|---|---|---|---|---|
| (S1) 0.5以上 | 毎日新聞記事（94年）1年分 | sim1 | wo格 | 0.5以上 | 50462 |
| (S1) 0.15以上 | 毎日新聞記事（94年）1年分 | sim1 | wo格 | 0.15以上 | 1775021 |
| (S1) 0.08以上 | 毎日新聞記事（94年）1年分 | sim1 | wo格 | 0.08以上 | 4805640 |
| (S2) 25以上 | 毎日新聞記事（94年）1年分 | sim2 | wo格 | 25以上 | 32100 |
| (S2) 15以上 | 毎日新聞記事（94年）1年分 | sim2 | wo格 | 15以上 | 165515 |
| (S2) 6以上 | 毎日新聞記事（94年）1年分 | sim2 | wo格 | 6以上 | 1891937 |
| (S1) 特許 | 特許179件（G06F1722） | sim1 | wo格、ga格 | 全て | 251199 |
| (S2) 特許 | 特許179件（G06F1722） | sim2 | wo格、ga格 | 全て | 251199 |
| (S2) 特許 L20以上 | 特許928件（G06F1722） | sim2 | wo格 | 20以上 | 1111 |
| (S2) 特許 L10以上 | 特許928件（G06F1722） | sim2 | wo格 | 10以上 | 27074 |
| (S2) 特許 L4以上 | 特許928件（G06F1722） | sim2 | wo格 | 4以上 | 1110650 |

特許文書からの自動獲得では、辞書作成のベースとなる単語区切りを、通常の単語単位（≒短単位：辞書に登録されている単語）とするほかに、合成語（≒長単位）ベースとした辞書も用意して実験した。
上の表で辞書名にLが入っているものが長単位ベースである。

## 1.5. 検索性能の評価式

本研究においては、検索性能の評価に、以下の評価値（評価式）を用いた。

① 正解順位の平均
　　正解順位の平均 ＝ 検索された正解の順位の和／検索された正解総数

② 再現率(recall)
　　再現率 ＝ 検索された正解総数／全正解数

③ 上位100位以内の再現率
　　上位１００位以内の再現率＝ 上位100位以内に検索された正解総数／全正解数

④ 精度（precision）
　　精度 ＝ 検索された正解総数／検索総数

⑤ 既検索正解の順位平均（拡張後の結果に対して）
　　既検索正解の順位平均差＝ 「拡張無しの正解順位の平均」－「拡張無しで検索された正解の拡張後の順位の平均」

今回の実験では、検索総数は最大1000件までとしている。

# 2.特許検索

## 2.1. 概要

特許検索の実験では、特許明細書の**請求項**」をクエリー文書とし、その特許に対して特許審査官が拒絶理由として示した**引例**を正解文書とした。諸元は次のとおりである。

| | | *備考* |
|---|---|---|
| 検索課題 | **特許引例** | |
| 検索対象 | **G06F1722** | *かな漢字変換* |
| 検索課題数 | **36** | |
| 検索対象数 | **615** | |
| 正解数 | **85** | |
| 検索語数（平均） | **35.61** | |
| 拡張語彙数 | **15** | *固定* |
| 辞書 | **EDR辞書** | *改良版* |
| | **自動獲得辞書** | *同分野の特許* |

## 2.2. EDR辞書利用の結果

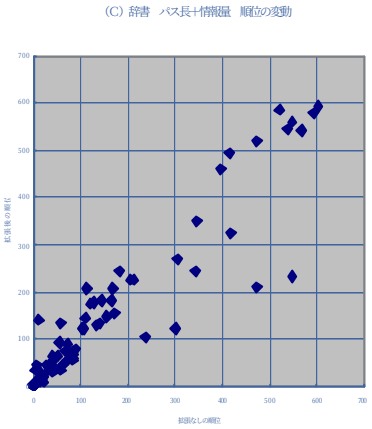
### 2.2.1. 正解順位の平均・再現率・精度

| **使用辞書** | **正解の順位の平均** | **検索総数** | **検索正解数** | **再現率** | **精度** | **上位100以内の再現率** |
|---|---|---|---|---|---|---|
| **拡張無し** | 134.9 | 22137 | 85 | 1.000 | 0.0038 | 0.61 |
| **EDR辞書V2 パス長** | 139.5 | 22410 | 85 | 1.000 | 0.0038 | |
| **(C) パス長** | 140.6 | 22137 | 85 | 1.000 | 0.0038 | |
| **(C) 情報量** | 135.2 | 22133 | 85 | 1.000 | 0.0038 | |
| **(C) パス長＋情報量** | 131.7 | 22134 | 85 | 1.000 | 0.0038 | 0.59 |
| **(i) パス長** | 145.7 | 22140 | 85 | 1.000 | 0.0038 | |
| **(ii) パス長** | 149.2 | 22137 | 85 | 1.000 | 0.0038 | |
| **(iv) パス長** | 141.0 | 22138 | 85 | 1.000 | 0.0038 | |
| **(i) 情報量** | 136.9 | 22133 | 85 | 1.000 | 0.0038 | |
| **(i) パス長＋情報量** | 138.8 | 22133 | 85 | 1.000 | 0.0038 | |

### 2.2.2. 分析

拡張なしの場合で、上位１００位以内の再現率が約６０%である。拡張なしの場合も含め、すべての検索で再現率が１となっている。すなわち検索漏れは起きていない。また検索総数もほとんど変化がなく、検索精度も変化はない。
語彙拡張を行った結果で、効果があらわれているのは、(C) 辞書でパス長ベースの類似度と情報量ベースの類似度の和を用いた場合で、正解の順位の平均が僅かではあるが良くなっている。ただし、上位１００位以内の再現率は向上していない。これ以外の結果では、いずれも語彙拡張により正解の順位の平均が僅かではあるが悪くなっている。

下は（C）辞書での語彙拡張における順位の変動を示したグラフである。




グラフを見ると、いくつかの検索文書で正解の順位が著しく向上していることがわかる。しかし１００位以内に入るにはいたっていない。また、逆に拡張なしで１００位以内に入っていた正解のいくつかが語彙拡張により１００位以下に下がってしまっている。これらの変化の相殺により、総合的な評価では検索性能がほとんど変化しないという結果となっていると考えられる。

## 2.3. 自動獲得辞書の結果

### 2.3.1. 正解順位の平均・再現率・精度

短単位ベースの場合

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 |
|---|---|---|---|---|---|
| 拡張無し | 134.9 | 22137 | 85 | 1.000 | 0.0038 |
| (S1) | 135.4 | 22139 | 85 | 1.000 | 0.0038 |
| (S2) | 145.1 | 22140 | 85 | 1.000 | 0.0038 |

長単位ベースの場合

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 |
|---|---|---|---|---|---|
| 拡張無し（L） | 167.7 | 22132 | 85 | 1.000 | 0.0038 |
| (S2)L20以上 | 167.0 | 22140 | 85 | 1.000 | 0.0038 |
| (S2)L10以上 | 165.3 | 22138 | 85 | 1.000 | 0.0038 |
| (S2)L4以上 | 169.0 | 22138 | 85 | 1.000 | 0.0038 |

### 2.3.2. 分析

短単位ベースの自動獲得辞書を用いた結果では、やはり検索性能に大きな変化は起きていない。
一方、長単位ベースの自動獲得辞書を用いた結果では、正解の順位の平均が僅かではあるが改善されていることがわかる。

## 2.4. 辞書の組み合わせの結果

### 2.4.1. 正解順位の平均 ･再現率 ･精度

短単位ベース

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 |
|---|---|---|---|---|---|
| 拡張無し | 134.9 | 22137 | 85 | 1.000 | 0.0038 |
| C)パス長＋情報量＋ S1) | 131.8 | 22134 | 85 | 1.000 | 0.0038 |
| C)パス長＋情報量＋ S2) | 131.9 | 22135 | 85 | 1.000 | 0.0038 |

長単位ベース

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 |
|---|---|---|---|---|---|
| 拡張無し (長単位) | 167.7 | 22132 | 85 | 1.000 | 0.0038 |
| C)パス長＋情報量+ (S2)L 10以上 | 160.2 | 22133 | 85 | 1.000 | 0.0038 |

### 2.4.2. 分析

EDR辞書と自動獲得辞書の併用を行っても、実験結果は大きくは変化していない。

# 3. 新聞記事検索

## 3.1. 検索課題の概要

検索対象は、日本語テストデータセットであるBMIR―J2を用いた。クエリーは、データセット中の検索要求文そのままを用い、正解はAランクの正解のみとした。（BMIR―J2の正解にはAランクとBランクがある。）検索語彙の拡張にあたっては、拡張語彙数は **15** に固定とした。以下に諸元を示す。

| | | 備考 |
|---|---|---|
| 検索課題 | **BMIR-J2** | |
| 検索対象 | **毎日新聞記事** | 1994年 |
| 検索課題数 | **50** | 追加分は除く |
| 検索対象数 | **5080** | |
| 正解数 | **665** | 正解A |
| 検索語数（平均） | **2.52** | |
| 拡張語彙数 | **15** | 固定 |

表でわかるように、検索語数の平均は3以下と、比較的短い検索課題が多い。
次に示すように、検索課題は機能によって6グループに分類されている。

| グループ | 説明 |
|---|---|
| A | 基本機能（キーワードの存在確認、キーワードのシソーラスによる展開語の存在確認、それらの語の存在に関する論理式の判定など）の |
| B | 数値・レンジ機能中心 |
| C | 構文解析（複数のキーワードのかかり受け関係の判断）機能中心 |
| D | 言語知識（通常の構文解析に必要とされるよりも深い言語知識）利用 |
| E | 世界知識（常識的な判断、蓄積された事実からの推論など）利用中心 |
| F | 言語知識と世界知識の併用 |

## 3.2. EDR辞書利用の結果

### 3.2.1. 正解順位の平均・再現率・精度

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 | 既検索正解順位平均差 |
|---|---|---|---|---|---|---|
| **拡張無し** | 82.6 | 20728 | 516 | 0.776 | 0.0249 | |
| **EDR辞書V2：パス長** | 104.3 | 32291 | 568 | 0.860 | 0.0157 | -11.7 |
| **(C) パス長** | 104.1 | 31442 | 554 | 0.833 | 0.0176 | -12.9 |
| **(C) 情報量** | 92.7 | 28043 | 564 | 0.848 | 0.0201 | 1.1 |
| **(C) パス長＋情報量** | 97.1 | 29277 | 562 | 0.845 | 0.0192 | -3.5 |
| **(I) パス長** | 107.6 | 34471 | 575 | 0.865 | 0.0167 | -21.8 |
| **(In) パス長** | 109.5 | 33456 | 575 | 0.865 | 0.0172 | |
| **(Iv) パス長** | 105.3 | 28284 | 536 | 0.806 | 0.0190 | |
| **(I) 情報量** | 89.2 | 28844 | 572 | 0.860 | 0.0198 | 3.8 |
| **(I) パス長＋情報量** | 92.0 | 30277 | 574 | 0.863 | 0.0190 | -0.2 |

### 3.2.2. 分析

無修正のＥＤＲ辞書を用いて、語彙拡張の実験を行い、拡張無しの場合と結果を比較した。この時、類似度としてはパス長ベースのものを用いた。検索された正解数は増えているが（再現率が約11%向上）、検索総数も増えた結果、精度は下がっている（約37%低下）。正解の平均の順位も３６%下がっている。

(C)辞書については、無修正のＥＤＲ辞書での結果と比較して、精度は少し改善されたが、検索される正解数が減少し、再現率の伸びが低下した。(T)辞書については、再現率は変化しなかったが、精度の改善は若干であった。

そこで、情報量ベースの類似度評価の実験を行ったところ、精度に大きな改善が認められた。(T)辞書に情報量ベースの場合、正解の順位の平均が８０位台を確保し、拡張無しの結果に比べ約８%の低下に抑えている。無修正のＥＤＲ辞書利用時と比べて再現率はほとんど低下せず、拡張無しの場合に比べて約11%向上している。検索精度も改善されて、約２０%の低下に抑えられている。

## 3.3. 自動獲得辞書利用の結果

### 3.3.1. 正解順位の平均・再現率・精度

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 | 既検索正解順位平均差 |
|---|---|---|---|---|---|---|
| 拡張無し | 82.6 | 20728 | 516 | 0.776 | 0.0249 | |
| (S1) 0.5以上 | 82.6 | 20834 | 516 | 0.776 | 0.0248 | |
| (S1) 0.15以上 | 87.2 | 33489 | 532 | 0.800 | 0.0159 | -0.3 |
| (S1) 0.08以上 | 85.4 | 33233 | 531 | 0.798 | 0.0160 | 0.2 |
| (S2) 25以上 | 89.8 | 41758 | 534 | 0.803 | 0.0128 | 1.6 |
| (S2) 15以上 | 93.0 | 44756 | 536 | 0.806 | 0.0120 | -0.4 |
| (S2) 6以上 | 99.7 | 46227 | 541 | 0.814 | 0.0117 | |

### 3.3.2. 分析

自動獲得辞書は、ＥＤＲ辞書に比べて再現率の伸びが低い。しかし正解の順位の平均は、ＥＤＲ辞書に比べると悪化していない。

また、(S2)辞書の方が再現率を伸ばす効果が大きく表れている。一方、検索精度という観点では、(S1)辞書の方が(S2)辞書より結果が良い。また、拡張無しの時点では検索漏れを起こしていた正解の順位が、(S1)辞書の方が高い。これは検索総数が少なく抑えられていることと関係している。

## 3.4. 辞書の組み合わせの結果

### 3.4.1. 正解順位の平均・再現率・精度

| 使用辞書 | 正解の順位の平均 | 検索総数 | 検索正解数 | 再現率 | 精度 | 既検索正解順位平均差 |
|---|---|---|---|---|---|---|
| 拡張無し | 82.6 | 20728 | 516 | 0.776 | 0.0249 | |
| (T)情報量＋(S1) 015以上 | 90.6 | 28710 | 572 | 0.860 | 0.0199 | 2.2 |
| (T)情報量＋(S2) 25以上 | 90.2 | 29729 | 581 | 0.874 | 0.0195 | 3.5 |

### 3.4.2. 分析

ＥＤＲ辞書と自動獲得辞書を組み合わせて用いることにより、単独で用いた時に比べて再現率がより向上している。しかも、精度や正解の順位の平均は、単独使用の時に比べてもあまり悪化していない。このことから、辞書の併用には効果があると考えることができる。

## 3.5. 効果が認められたケース

### 3.5.1. 検索漏れにヒットしたケース

「拡張無し」では検索されず、拡張後には（拡張無しの正解平均順位である）82位以内に検索された正解がある課題は次のものである。

| 辞書 | 課題番号 |
|---|---|
| ①パス長 | 105 107 108 110 112 113 123 140 |
| ①情報量 | 107 108 110 112 113 127 131 140 |
| ⑤1) 15以上 | 105 108 110 |
| ⑤2) 25以上 | 110 |

この結果を見ると、検索漏れとなっていた記事をより精度よく検索することを目的とすると、ＥＤＲ辞書を用いた語彙拡張の方が、コーパス学習辞書を用いた語彙拡張より、効果が得られているケースが多いことがわかる。また、検索課題のグループ別の観点からは、検索漏れとなっていた記事を検索する効果は、グループ (A) でもっとも大きくなっている。

これらのケースでは、同義語への語彙拡張が効果を発揮している場合と、上位語から下位語への語彙拡張が効果を発揮している場合が見られる。前者の例としては課題 107、113 などがある。課題 107 では「ビデオデッキ」に対して「ビデオ」、課題 113 では「ソフトウエア」に対して「ソフト」という拡張語が効果を発揮している。後者の例としては、課題105、112などがある。課題105 では「飲料」に対して「ビール」、課題112 では「核兵器」に対して「原爆」「原水爆」などが効果を発揮している。

### 3.5.2. 正解の順位が向上したケース

次に、「拡張無し」で検索された順位より５位以上順位が上がり、かつ８２位以内に検索された正解がある課題は、次のものである。

| 辞書 | 課題番号 |
|---|---|
| ①パス長 | 104 106 108 112 115 116 117 119 120 123 125 126 128 130 131 132 134 135 137 139 144 150 |
| ①情報量 | 106 108 109 112 119 123 128 130 131 132 134 135 137 139 141 144 145 148 150 |
| ⑤1) 15以上 | 102 104 108 109 112 115 116 120 128 129 134 135 137 138 144 145 150 |
| ⑤2) 25以上 | 109 112 117 120 124 127 129 130 137 139 140 144 145 150 |

課題 132 のように上位（「コンピュータ」）から下位（「パソコン」）への言い換えが効果を発揮している例もあるが、むしろ類義表現（言い換え表現）への拡張が効果を発揮している例が目に付く。例えば、課題 131 では「株価」に対して「相場」「終値」など、また「動向」に対して「流れ」「行方」などが効果を発揮している。同様に、課題 137 では「映画」に対して「アニメ」「ビデオ」「スクリーン」などの単語が効果を発揮している。

## 3.6. 検索漏れを検索できなかったケース

拡張無しでも拡張有りでも検索されなかった正解のある課題で、拡張無しで検索されなかった正解の全てが、拡張後も検索されなかった課題は次のものである。

| 辞書 | 課題番号 |
|---|---|
| ①パス長 | 102 104 111 124 127 129 134 138 147 |
| ①情報量 | 102 104 105 111 124 134 138 141 147 |
| ⑤1) 15以上 | 102 104 107 111 113 129 134 140 141 143 144 146 147 |
| ⑤2) 25以上 | 102 107 108 113 127 128 129 131 134 135 138 139 141 143 147 |

この中で、4 つの辞書に共通する課題がある。102、147 である。
102 は、『固有名詞への展開』が必要であったケースと考えられる。検索されなかったのは、「航空３社」のうちの「全日空」のみが話題となっている記事であった。「全日空」のみの記事でかつ普通名詞の「航空」が使用されていない記事では、固有名詞への展開がないために検索漏れになっていると考えられる。
課題 147 は、検索文中の単語「トップ」「不況」「対策」「発言」がそのままの表現では正解記事に含まれず、かつ語彙拡張によっても正解記事中の表現に辿り着けなかったというケースである。
ＥＤＲ辞書では、「トップ」は任意の組織中の“長”を指している。ＥＤＲ辞書にも「トップ」にそのような語義は存在し、類義語も多数あるが、その数は拡張語数の制限内に収まらないほど多く、その中で今回の記事に含まれる特定の語彙（例えば「首相」「社長」など）が拡張語に含まれなかった。次に「不況」であるが、正解記事は「不況」について直接述べているわけではない。「対策」についても、具体的な“不況対策”として「公定歩合の引き上げ」「減税」などを述べている場合、対策であるということを述べているわけではない。また「発言」については、「会見」等の単語への拡張が必要であったが、EDR辞書によってもまた自動獲得辞書によっても、このような拡張は行えなかった。
一方、自動獲得辞書では、「トップ」と「首相」、「発言」と「会見」、「対策」と「減税」のいずれもが獲得されて辞書に登録されていた。ところが、これらの単語は多数の類義語が獲得されており、その中で、これらの拡張語は上位１５位内に残ることができなかった。
次に、107、113、143 には、共通する問題点があった。それは、検索文中のインデックス語（「ビデオデッキ」、「ソフトウエア」、「経営陣」、「刷新」）が学習対象（⇐検索対象でもある）のコーパスの中にほとんど出現していなかったという点である。すなわち、このケースでは、検索語がオントロジー辞書にいわば“未登録”という状態になっていた。

## 3.7. 副作用

拡張無しの時に２０位以内であった正解で、拡張により１０位以上順位を下げた正解がある課題を取り出してみると、次ようになる。

| 辞書 | 課題番号 |
|---|---|
| ①パス長 | 102 104 105 108 109 112 115 116 120 122 123 125 132 134 137 140 144 146 |
| ①情報量 | 104 108 112 123 134 137 140 144 |
| ⑤1) 15以上 | 109 115 140 |
| ⑤2) 25以上 | 109 |

これらは、拡張語彙に含まれていた不適切な語彙により、不正解の記事が順位を上げたために、相対的に正解の順位が下がったものと考えられる。

例えば、課題104の場合（(I)情報量利用）、農薬」の拡張語の中に 枯れ葉剤」という単語が含まれており、主にこの単語が不正解の記事の順位を上げている。課題 108 の場合、携帯電話」に対して 電話」という上位にあたる拡張語が含まれていたため、この単語が悪影響を及ぼしたものと考えられる。また、課題 109 の場合、減税」という単語に対し自動獲得した辞書には 支援」改正」などの関連語が多く登録され、これらが関連性の無い記事の順位を引き上げてしまったと考えられる。
この副作用は、EDR辞書を利用した語彙拡張の場合により多く発生し、自動獲得辞書を利用した辞書の方が副作用は少ないことがわかる。
また、副作用は「(I)パス長」で特に大きく、「(I)情報量」の方が少なくなっている。このことは、情報量ベースでの評価値の採用が、パス長ベースで評価した際に起きる副作用を抑える効果を発揮したことを示している。

# 4. 結論

## 4.1. EDR辞書の利用

1. 新聞記事検索（BMIR－J2）において、改良版EDR辞書による語彙拡張の有効性を確認した。

   無修正のEDR辞書を利用した場合
   再現率１１%向上 精度３７%低下 既検索の正解の平均順位１４%低下

   修正後のＥＤＲ辞書（トリミング&情報量ベース類似度）を利用した場合
   再現率１１%向上 精度２０%低下 既検索の正解の平均順位5%向上

   特に、概念の出現頻度の情報を加味した情報量ベース類似度の効果が大きく表れている。

2. 効果ありの要因

   - 検索漏れにヒット
     同義語への語彙拡張が効果を発揮している場合と、上位語から下位語への語彙拡張が効果を発揮している場合とがある。特に課題グループAに対して効果が大である。

   - 正解の順位の向上
     類義表現（言い換え表現）への拡張が効果を発揮している場合が目に付く。

3. 副作用の要因

   - 不正解文書へのヒット
     上位語への語彙拡張が副作用を起こしている場合
     不適切な下位語への語彙拡張が副作用を起こしている場合
     関連語への語彙拡張が副作用を起こしている場合

## 4.2. オントロジーの自動獲得

1. 新聞記事検索（BMIR－J2）において若干の効果を確認した。

   再現率3%向上 精度４８%低下 既検索の正解の平均順位2%向上

2. 効果が発揮されなかった要因

   - 検索語彙が検索対象文書にほとんと出現しない場合
     検索対象文書をコーパスとした自動獲得では、出現しない語彙の情報は獲得できない。

- 固有名詞への語彙拡張が必要な場合
  今回の自動獲得では固有名詞を対象に含めなかったため、普通名詞から固有名詞への語彙拡張が必要とされるような検索課題では効果が発揮されなかった・

## 4.3. 課題による効果のばらつき

新聞記事検索と特許検索では、語彙拡張による効果に大きな違いが出た。また、BMIR-J2の検索でも課題グループにより効果の出方に違いがみられた。これらの要因をまとめると次のようになる。

- 課題が複雑になりすぎると、語彙拡張の効果が打ち消される。特に検索文が複数の内容から構成されている場合、語彙拡張の効果が現れない。
- 検索課題が単純な場合、語彙拡張により、検索漏れしていた文書にヒットする効果が大きい。この場合、同義語、下位語への語彙拡張が効果を発揮するケースが多い。
- 検索課題が複雑な場合、語彙拡張により、正解の順位が向上する効果が大きい。この場合、関連語への語彙拡張が効果を発揮するケースが目に付く。

# 5. 今後の予定

## 5.1. 汎用オントロジー技術

### 5.1.1. 必要な情報を弁別的に提供できる構成への見直し

検索課題の性質によって、同義語／下位語への語彙拡張が効果を発揮する場合と、関連語への語彙拡張が効果を発揮する場合がある。オントロジー辞書の構成としてこれらの情報をさらに弁別的に提供できることにより、このような要求に応えることができる。

### 5.1.2. 概念の頻度情報の利用／獲得方法の検討

頻度情報を用いた類似度評価式の改良により語彙拡張の効果が改善され、オントロジー辞書には概念間の関係の記述のみではなく、各々の概念の出現頻度の情報も重要な要素であることが確認された。さらに検討すべき課題である。

## 5.2. オントロジー生成技術

### 5.2.1. 意味関係の明確化技術の開発

自動獲得したオントロジー知識をより詳細化し、意味関係を明確化するための技術の開発を検討する。

### 5.2.2. 固有名詞に関する知識の獲得

このような知識の一部は汎用オントロジーとしても登録すべきものであるが、自動獲得の重要な要素のひとつとなると考えられる。いわゆる 情報抽出」研究の分野で用いられているような技術の導入を検討する必要がある。

## 5.3. 検索方式の改良

特許検索においては、検索文中に複数の内容が存在することにより、語彙拡張の効果を打ち消していると推測される。この問題に対する一つ解決方法として、ブール式検索の併用が考えられる。

## 5.4. 類義表現／言い換えの研究

特許検索において語彙拡張の効果が発揮されなかった原因のひとつとして、語彙の拡張だけでは同義表現・類義表現を認識する能力に不足していたことが考えられる。言い換え(paraphrase)の研究は今後の発展が期待されており、知見を蓄積していく必要がある。

# あとがき

オントロジーを用いた概念検索システム、オントロジー生成システムをそれぞれ開発し、これを用いて特許文書、ならびに日本語検索テストデータセット（BMIR－J２）において、検索語彙拡張の実験を実施した。
その結果、BMIR－J2において、汎用オントロジー技術により再現率が約１１%向上することを確認した。またオントロジー生成技術により再現率が約３%向上することを確認した。（いずれも、語彙拡張前と比べ、正解文書の順位の平均値も低下していないことを確認済み。）
今後は、オントロジー知識をより詳細化し、意味関係を明確化するための技術の開発を検討する。また複雑な検索課題での効果を高めるため、言い換え(paraphrase)の知見の蓄積を図る。

# 成果発表、特許等の状況

- 研究発表

| 著者名 | 論文表題 | | |
|---|---|---|---|
| 斎藤佳美、大嶽能久、木村和広 | 日本語文書検索における、シソーラスによるクエリー拡張効果の分析 | | |
| 雑誌名 | 巻 | 発行年 | ページ |
| 信学技報 | NLC 2001-7 (2001-5) | 2001 年 | 41-48 |

# 参考文献

EDR (1995). "EDR 電子化辞書仕様説明書(第 2 版)." EDR Technical Report TR-045.

Kimura, K. and Hirakawa, H. (2000). "Abstraction of the EDR Concept Classification and its Effectiveness in Word Sense Disambiguation." In Proceedings of the 2nd International Conference on Language Resources and Evaluation(LREC-2000), pp.615-622.

Mandala, R., Tokunaga, T., and Tanaka, H.. (2000). "Using Multiple Thesaurus Types for Query Expansion.." Journal of Natural Language Processing, 7(2), pp.117-140.

Miller, G.(1990). "WordNet：An On-line Lexical Database." Special Issue of International Journal of Lexicography, 3(4).

Resnik, P. (1995). "Using Information Content to Evaluate Semantic Similarity in a Taxonomy." In Proceedings of the 14th International Joint Conference on Artificial Intelligence(IJCAI-95), pp.448-453.

.