社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－NL－176 (13)
2006／11／23

文脈情報による同義語辞書作成支援ツール

寺田昭† 吉田稔‡ 中川裕志‡

†(株) 日本航空インターナショナル
‡東京大学情報基盤センター

あらまし 情報検索，テキストマイニングなどのテキスト処理の効率を向上させるには，同義語辞書の作成が必要である．航空などの分野では，漢字/ひらがなだけでなく，カタカナ，アルファベット，およびそれらの略語が同義語として用いられる．常に新しい語が発生するので，辞書の定期的な更新が必要となるが，それを人手で行うのは大変な作業である．本論文では，同義語辞書作成を半自動化するツールを提案する．システムは，クエリが与えられると意味的に同じ語の候補を提示する．辞書作成者は，その中から同義語を選択して，辞書登録を行うことができる．類似度は，前後に出現する単語の文脈情報により計算した．航空分野でのパイロットレポートを使用して実験を行い，その評価に平均精度を用いた．

キーワード：同義語 文脈情報 辞書作成

A Tool for Constructing a Synonym Dictionary using Context Information

Akira Terada† Minoru Yoshida‡ Hiroshi Nakagawa‡

†Japan Airlines International Co.,Ltd.
‡Information Technology Center, The University of Tokyo

Abstract To improve the proficiency of text processing such as information retrieval or text mining, it is necessary to construct a synonym dictionary, but it is very tiresome to make it by hands. In some fields, such as aviation, synonym nouns are mixed with kanji/hiragana, katakana, alphabet and their abbreviations. As new words always come to be used, the dictionary update is a big issue. In this paper, we propose a tool for constructing a synonym dictionary. The system will return synonym candidates against a query. A synonym can be easily registered in dictionay by looking the synonym candidates. We experimented the system performance by aviation pilot report and evaluated it by average precision.

Keywords : *synonym* *context information* *constructing a dictionary*

1 はじめに

情報検索，テキストマイニング，情報抽出などのテキスト処理を行う場合，同義語の処理が問題となる．同義語の同定を行わないと，処理能力が低下してしまう．例えば，「鳥」を含む文書を検索したい場合，「鳥」と「Bird」が同定できなければ，検索語として「鳥」を指定しただけでは，「Bird」を含み「鳥」を含まない文書は検索できない．

航空分野のマニュアル，補足情報，業務報告書等に使用される名詞は，漢字/ひらがなだけでなく，カタカナ，アルファベット，そして，それらの略語が混じったものが使用される場合が多い．例えば，飛行機のマニュアルの場合，「Flap」を「高揚力装置」と表現するよりも，「Flap」と表現し，用語の使用がマニュアルよりも自由なマニュアル以外のテキストでは，「Flap」や「フ

ラップ」と表現している．その理由は，海外から輸入された語句は，漢字で表現するとイメージがつかみ難いものがあるためであり，そのような語句は，英語表現や英語のカタカナ表現が使用される．「Aileron」を「補助翼」というよりは，「Aileron」や「エルロン」と通常表現している．マニュアルの場合は，ある程度，使用語が統一されているが，マニュアル以外のテキストは，語句の使用がより自由で分野の異なる人間や計算機にとって理解し難いものとなっている．例えば，"滑走路"を略語を使用して"RWY"，"R/W"と表現している．このようなテキストを計算機で処理する場合には，同義語辞書が必要であるが，これらの語句は汎用の辞書に載っていない場合が多い．さらに，語句の使用は，統制されているものではなく，また，常に新しい語が使用されるので，一度，分野の辞書を作成しても，それを定期的にメンテナンスする必要がある．これを人手だけで行うのは大変な作業である．

本論文では，同義語辞書の作成を計算機が支援するツールを提案する．計算機は，与えられたクエリに対して，意味的に同じ語（同義語）の候補を提示する．辞書作成者は，クエリをシステムに与えることにより，同義語の候補語をシステムから提示され，その中から同義語を選択して，辞書登録をすることができる．

実験は，航空分野でのパイロットレポートを使用した．この分野では，上の例のように汎用の辞書に載っていない語が多く存在する．

評価は，平均精度を用いて行い，他の手法と比較して満足できる結果が得られた．

論文構成は，第2節では関連研究について，第3節では類似度の計算アルゴリズムについて，第4節では航空分野でのパイロットレポートを用いた実験について述べる．最初に，単名詞についての処理を述べ，複合名詞については，専門用語自動抽出システム [10] が抽出した複合名詞を使用することにより単名詞と同様の手法を用いた．第5節では辞書の作成に考察する．第6節では結論と今後の研究課題について述べる．

2 関連研究

同義語を自動的に計算する研究は，これまで数多く行われてきた．カタカナと英語の対応，英語とその略語の対応，日本語とその略語の対応などがある．略語処理では，略語の近傍に括弧書きで略語の定義がされている場合の研究がある [1],[6]．この手法は，略語の定義が略語の近傍でされているものについては有効であるが，必ずしも略語の定義がされているとは限らない．本論文で扱う文書にも略語の定義はされていないので，この手法は適用できない．カタカナと英語の対応では，Knight らは，カタカナと英語の対応を発音記号から対応付けしている [2]．阿玉らは，カタカナのローマ字表記と英語との対応付けをしている [9]．Terada らは，英語とその略語の対応を両者に含まれる文字及びその順序が等しいなどの情報を使用することで同定している [7]．

同義語とその候補語の類似度の計算は，文脈情報から余弦を用いて計算するものが多い．文脈情報として，語の前後の局所的なものを用いるもの [7]，文書全体から抽出して用いるものがある [8]．Ohtake らは，カタカナの変形を探すのに，エディット距離で候補を絞った後に，文脈情報を用いているが，その際，カタカナが用いられている構文を解析して，動詞，名詞，助詞を使用している [5]．Masuyama らは，カタカナ処理で WEB データから英語に対応するカタカナのエディット情報を取得している [4]．

計算量の削減及び精度の向上のために，文字情報を用いて，対応関係を絞り込む，または，決定する研究が多い．文脈情報を用いる場合には，全ての語を用いるのでなく，内容語を用いるものが多い．

本論文では，漢字，ひらがな，カタカナ，アルファベット，略語の類似度を同時に計算するために，文字情報による絞り込みは行わず，文脈情報のみでどの程度の精度が得られるかを実験した．

3 類似度の計算

クエリに対してシステムは，同義語候補語の中から順位付けをして，出力する．本節では，同義語候補語，文脈情報について説明し，類似度の計算アルゴリズムについて説明する．最初に単名詞の処理について述べ，複合名詞についての処理については，4.7節で述べる．

3.1 同義語候補語

単名詞の同義語候補語として，形態素解析器が出力したアルファベット，名詞，カタカナとした．形態素解析器としては，茶筅[1]を使用し，その中で出現頻度が，100以上のものを使用した．

3.2 文脈情報

「同義語は，同じような文脈で使用される」という仮定から，文脈情報により語の類似度を計算できると考えた．これは，人間が語の意味を理解するのにその

[1] http://chasen.naist.jp/hiki/ChaSen/

語が出現する前後の文脈から類推しているという仮定からである．文脈は，同義語の近傍の語 (局所的文脈) が影響していると仮定した．人間は，前後の語の中で，場面に応じて文脈語を選別をしていると考えられるが，計算機で実現するのは不可能であるので，場面に応じた選別はしていない．クエリを q とし，その前後の語の並びを，$x_\alpha ... x_2 x_1 q y_1 y_2 ... y_\beta$ とする．前後の語は，形態素解析器が出力した語とする．対象とするクエリの文脈語をクエリの前で $x_\alpha ... x_1$，クエリの後ろで $y_1 ... y_\beta$ とすると，window 幅は α，β で，window[α, β] と表すこととする．同義語候補語の window 幅についても，同様の表現とする．window 幅は，同義語，同義語候補語を含む 1 文の範囲内だけを考慮した．文脈語の選択については，次節で述べる．

3.3 類似度

クエリ (query) の文脈情報を $\boldsymbol{c_q}$，同義語候補語 (synonym) の文脈情報を $\boldsymbol{c_s}$ とする．$\boldsymbol{c_q}$ と $\boldsymbol{c_s}$ をベクトル空間モデルで表し，その類似度をベクトルの余弦で表すと，クエリと同義語候補語の類似度 (sim) は，次式で計算される．

$$sim(query, synonym) = \frac{\mathbf{c_q} \cdot \mathbf{c_s}}{|\mathbf{c_q}| \cdot |\mathbf{c_s}|} \quad (1)$$

3.4 平均精度

情報検索の性能評価として精度と再現率がよく用いられるが，これらは，与えられたクエリに対する検索結果全体に対する性能を表すものである．検索結果から辞書作成者が辞書登録することを考えると，検索結果の順位に於ける精度が重要である．つまり，上位の検索結果ほど評価値は，高い必要がある．したがって，このような評価尺度を表すものとして平均精度 (average precision) を用いた．N 個のクエリの評価をする場合，i 番目のクエリに対する平均精度は以下で表される:

$$AveragePrecision[i] = \frac{1}{R[i]} \sum_{j=1}^{n} (rel[j] \cdot \sum_{k=1}^{j} rel[k]/j) \quad (2)$$

ここで、
n：同義語の候補数
R[i]：i 番目のクエリの同義語数
rel[k]：システムが順序付けした回答の中で，k 番目の回答が正解であれば 1，そうでなければ 0
i 番目のクエリに対する平均精度は，検索結果の正解の場所の精度 $\sum_{k=1}^{j} rel[k]/j$ の同義語 i 番目全体に対する和を同義語数 R[i] で割ったものである．

N 個のクエリ全体の平均精度は，個々のクエリに対する平均精度の平均で表される：

$$AveragePrecision = \frac{1}{N} \sum_{i=1}^{N} AveragePrecision[i] \quad (3)$$

4 実験

4.1 コーパス

コーパスとして，航空分野でのパイロットレポートを使用した．このレポートからは，事前に名前等の個人情報は削除し，個人を特定できないようにしてある．レポートの内容には，出発地・到着地などの定型情報とテキストで自由に記述された表題，本文が含まれているが，本論文では，本文を対象とした．まず最初に単単語についての処理について述べる．1,992 年から 2,003 年までの 6,427 件のレポートの本文のサイズは，約 6.9M バイトであった．同義語候補語は，茶筅で形態素解析した単語の名詞 (漢字/ひらがな)，アルファベット，カタカナを対象とし，その出現頻度が 100 以上のものを対象とした．その結果，同義語候補語の数は，1,343 になった．同義語抽出のタスクは，クエリの同義語をこれらの同義語候補語の中から選択するものとした．

4.2 評価用辞書

今回の実験評価のために，4.1 と同じ条件で同義語辞書を人手で作成した．その結果，辞書の登録数は 406，同義語数は 777 で，平均同義語数は 1.91 であった．同義語の中には，「Service」，「SVC」，「サービス」のように英語とその略語およびそのカタカナ表現のほか，「Traffic」，「相手機」のようにドメイン特有のものも含まれる．

4.3 文脈語の重み付けによる比較

クエリと同義語候補の文脈語としてそれぞれの前後に出現する内容語とし，名詞，アルファベット，カタカナ，動詞，形容詞を使用した．文脈情報は，文脈語の頻度ベクトルとした．

類似度は，3.3 節で述べたように余弦で計算する．クエリの文脈ベクトルを $\boldsymbol{c_q}$=(q_1,...,q_N)，同義語候補語の

表 1: 文脈語の頻度の log による補正の平均精度 (%) への影響の比較

	window [2,2]	window [3,3]
log による補正なし	27.3	28.3
log による補正あり	43.1	39.2

文脈ベクトルを c_s=(s_1,...,s_N) とすると，類似度 (sim) は次式で表される．

$$sim(query, synonym) = \frac{\mathbf{c_q} \cdot \mathbf{c_s}}{|\mathbf{c_q}| \cdot |\mathbf{c_s}|} = \frac{\sum_{i=1}^{N} q_i s_i}{\sqrt{\sum_{i=1}^{N} q_i^2 \sum_{i=1}^{N} s_i^2}} \quad (4)$$

ここで，文脈ベクトルの各要素 (x_i=q_i 又は s_i) は，頻度を表す．頻度 (x_i) は，通常 $\log(x_i + 1)$ のように調整して使用される場合が多い [3]．文脈ベクトルの各要素 (x_i) に対して，x_i を $\log(x_i + 1)$ として補正したものと，補正しないものを window [2,2] および window [3,3] で比較した．ここで，文脈語の頻度は，最小頻度 50，最大頻度 600 のものを用いた．結果は，頻度 (x_i) を $\log(x_i+1)$ で補正したものの方が window [2,2] では，約 16%，window [3,3] では約 11%よくなった (表 1)．この結果，文脈語の頻度の log による補正が，平均精度に与える影響が大きいことが確認できた．これ以降の実験では，文脈語の頻度を log により補正したものを用いることとする．

文脈語の $tf \cdot idf$ 的な補正について調べた．文脈語ベクトルが同義語候補語 (N) に対して 0 でない同義語候補語の数を df とすると，idf は，以下の式で計算される．

$$idf = log\frac{N}{df} + 1 \quad (5)$$

tf に対して，$tf \cdot idf$ を使用することによる効果を調べた．window[2,2] で文脈語の最小頻度 50，最大頻度 600 のものについて $tf \cdot idf$ の補正をしたものとしないものを比較したが，平均精度はどちらも 43.1%であった．window[3,3] で文脈語の最小頻度 50，最大頻度 600 のものについての比較でも平均精度はどちらも 39.2%であった．最小頻度を 50 としていることが，idf 的な補正をしているのと同等の効果がある可能性を考慮し，最小頻度 5 のものについて比較した．その結果，$tf \cdot idf$ の補正なしが 44.5%，補正ありが 43.1%で，$tf \cdot idf$ の補正はしない方がよい結果となった．

4.4 文脈語の選択による比較

文脈語の頻度による選択および品詞による選択の平均精度への影響について調べた．文脈語の最小頻度の平均精度への影響を図 1 に示す．最小頻度の設定は，それ以下の文脈語を含めるとノイズのために性能が低下するとの仮定からであるが，本実験では，最大頻度を 600 に固定して最小頻度を変化させると最小頻度が 5 あたりで最も平均精度が高かった．しかしながら，文脈語の最小頻度を小さくすると，文脈語数が増加し計算量が増加する欠点がある．

文脈語の最大頻度の平均精度への影響を図 2 に示す．本実験では，最小度を 50 に固定して最大頻度を変化させると最大頻度が 600 あたりで最も平均精度が高かった．

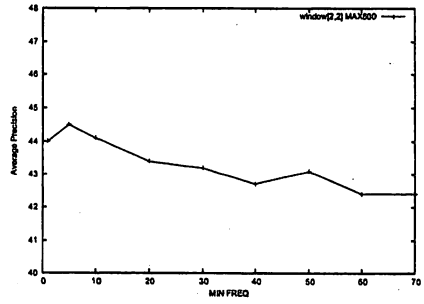

図 1: 文脈語の最小頻度による平均精度への影響

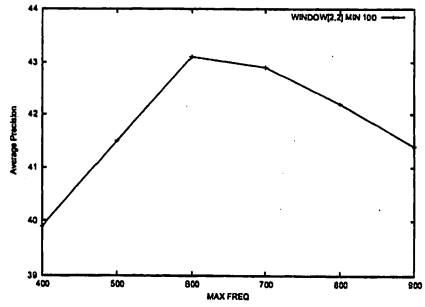

図 2: 文脈語の最大頻度による平均精度への影響

同義語を構成する名詞，アルファベット，カタカナを基本として，それに動詞を加えたもの，動詞，形容詞を加えたもの，動詞，形容詞，助詞を加えたものについて平均精度への影響を window[2,2] で実験した．なお，文脈語は，最小頻度が 50，最大頻度が 600 のものを使用した (表 2)．名詞，アルファベット，カタカナに対して，動詞，形容詞を加えたものが平均精度が一番よくなった．これは，同義語の前後でよりその語を判別できる情報が増加したからだと考えられる．助詞を加えたものについて，平均精度が下がったのは，助詞は，様々な語と共起するので，判別能力が下がったものと考え

表 2: 文脈語の選択による比較 (window [2,2])

	平均精度 (%)
名詞，アルファベット，カタカナ	40.5
名詞，アルファベット，カタカナ，動詞	42.3
名詞，アルファベット，カタカナ，動詞，形容詞	43.1
名詞，アルファベット，カタカナ，動詞，形容詞，助詞	27.3

られる．

4.5 window 幅による比較

window 幅による平均精度の比較を文脈語の最小頻度を 50，最大頻度を 600 にして行った．window 幅を同義語候補語の前 (FWD) に 0〜4 語，後 (AFT) に 0〜4 語変化させて実験した．その結果を表 3 に示す．平均精度は，window[2,2] が 43.1%で最もよかった．

同義語候補語の前後の window の比較では，window[2,0] では 35.5%，window[0,2] では 37.8%であった．例えば，「Boarding」というクエリに対する正解は「搭乗」であるが，window[2,0] では「搭乗」が 1 位になるが，window[0,2] では 8 位になる．理由としては，window[2,0] では「Boarding」と「搭乗」の前に共通の語である「お客様」が多く出現するが，window[0,2] では「Boarding」と「搭乗」の後に共通の文字列 (例:「を開始」など) の出現が少ないためであると考えられる．window[2,2] では，window[2,0] の影響を受けて 1 位になっている．「CAT[2]」というクエリに対する正解は「TURB」と「揺れ」であるが，window[2,0] では「TURB」が 2 位，「揺れ」が 9 位になる．共通に出現する代表的な言葉は「突然の」であるが，その数がそれ程多くない．window[0,2] では「TURB」が 1 位，「揺れ」が 2 位になる．その理由として，後に「に遭遇」という表現が多く出現しているからだと考えられる．window[2,2] では，window[0,2] の影響を受けて「TURB」が 1 位，「揺れ」が 2 位になっている．window[2,2] では，同義語の前 2 語の window[2,0] と同義語の後 2 語の window[0,2] が補完しあって，よい結果になっているものと考えられる．文脈語の window 幅における比較では，同義語候補語の window 幅の前後の選択による平均精度の顕著な差は認められなかった．

[2]Clear Air Turbulence

表 3: window 幅による平均精度 (%) の比較

FWD \AFT	0	1	2	3	4
0	-	21.4	37.8	37.1	33.0
1	25.6	31.5	40.7	38.7	35.3
2	35.5	36.9	43.1	40.1	36.8
3	34.8	37.8	40.9	39.2	36.1
4	32.2	34.6	37.4	36.3	33.3

表 4: 大域的文脈情報と局所的文脈情報の比較

	平均精度 (%)
酒井らの方式	7.4
window 22	39.5

4.6 大域的文脈情報との比較

酒井らは，日本語の略語からその原型語の対応関係を取得するのに以下のような手法を用いている [8]．略語候補とそれに対応する原型語の候補を，それを構成している文字情報から獲得する．略語候補と原型語の候補の類似度を計算して，対応関係を取得する．本論文では，文脈情報の類似度について我々の提案手法との比較を行った．彼らは，名詞の略語を対象としたが，名詞，カタカナ，ローマ字に拡張して我々の提案との比較を行った．類似度の計算は，コーパス中の略語候補語を含んでいる文書における略語候補語の出現頻度，全ての名詞の総出現頻度，文の数，略語候補語が最初に出現する文番号の情報を用いて重みを付与して順位付けを行い，その上位 n 文書を取り出して，略語候補の関連文書としている．次に，その関連文書に含まれる各名詞に対して出現頻度，文書頻度などの情報を用いて重みを付与して順位付けを行い，上位 m 個の名詞を取り出し，名詞の重みを付与したベクトルを生成している．原型語候補に対しても同様のベクトルを生成する．そして，その余弦により類似度を計算している．酒井ら [8] と同様に，n=20，m=200 として実験した．

結果は，表 4 にあるように，提案手法よりも，かなり低い値となった．その原因として，略語とその原型語の対応関係を求めるのに，関連文書全体から代表的な名詞を抽出して類似度を計算している (大域的文脈情報) が，必ずしも，略語に関連する文書があるとは限らないと考えられる．我々は，局所的な文脈語から類似計算を行っている (局所的文脈情報) が，この手法の優秀性が証明された．

4.7 複合名詞の処理

複合名詞については，専門用語抽出システム [10] が抽出したものの中で，重要度評価値が 3,000 以上の用語の中の複合名詞を選択した．専門用語抽出システムは，単名詞の左右に出現する単名詞の連接種類数と連接頻度および候補語の出現頻度から専門用語を抽出するものである．その結果，350 の複合名詞が得られた．人手で複合名詞に対して同義語辞書を作成した．同義語の中には複合名詞も単名詞も含まれる．その結果，辞書の登録数 73 で平均同義語数は，2.00 であった．この複合名詞 350 と単名詞 1,343 に対して window [2,2] で文脈語の最小頻度 50，最大頻度 600 で文脈情報を取得した．実験の結果，平均精度は，44.3%であった．辞書登録数が少ないので単純には比較できないものの単名詞間と同等の平均精度が得られた．

複合名詞と単名詞について以下のような関係があることが分かった．

1. 複合名詞の同義語が単名詞の同義語の組み合わせでできているもの：例：出発 遅れ - 出発 遅延
2. 複合名詞の基底名詞と単名詞が同義なもの：
   例：搭乗 券 - 券
3. 複合名詞の基底名詞以外の語同士が同義なもの：
   例：整備 点検 - 整備
4. 複合名詞の中で一部の名詞に省略があるもの：
   例：搭乗 旅客 数 - 搭乗 数
5. 単名詞同士では，同義でなかったものが複合名詞では同義になるもの：
   例：搭乗 口 - ゲート，到着 地 - 目的 地

1については、単名詞の同義語を置き換えることにより複合名詞の同義語を得ることが可能であるが，その場合には，「DEP 遅延」のようにあまり使用されない複合名詞の同義語が得られてしまい，単名詞の同義語を置き換えだけでは複合名詞の同義語を絞り込むことができない．2と3については，複合名詞を構成する名詞の中でより一般的で省略しても意味が変化しないものが省略されている．5の関係は，上記4種類と異なり，単純に省略や単名詞の置き換えだけでは扱えないもので，複合名詞の処理を行わないと同義語が得られないものである．

5 同義語辞書作成

同義語辞書は，表5のように見出し語に対して1語以上の同義語が辞書項目として登録される．情報検索やテキストマイニングでは，同じ概念をグループ化し精度を向上させるために見出し語に対して1対1で同義語を対応させる必要がある場合がある．例えば，表5に対して，同義語リストは表6のようになる．表6では，「APP」が「進入」に，「Approach」が「進入」に，「CRZ」が「巡航」に変換されることを示す．「進入」，「巡航」は，変換されないでそのまま使用される．複数の同義語の中からどの語を変換語に選択するかは，専門用語抽出システムの重要度評価値の最も大きなものを用いた．つまり，同義語同士の中で最も重要度の高い語に変換するものである．もちろん多義性のある語では，一意に同義語を決定できないのでこのようなリストは使用できない．この場合には，個々の語が出ている文脈から判断する必要があるが，これは今後の課題である．

次に同義語辞書を作成する際に一度に全て作成するのではなく，同義語辞書を以下に示すように一部作成した段階で同義語リストを文脈情報の正規化に使用するためにシステムに与えることにより，残りの辞書作成の精度（平均精度）が向上するかを検証した．例えば，「PAX」を「旅客」に変換することにより，「PAX Boarding」と「旅客搭乗」の「Boarding」と「搭乗」の文脈語が同一になる．文脈語としては，出現頻度が 100 以上 600 以下のものを用いているので，出現頻度が 500 以上の同義語リストを作成したところ，42 個の同義語リストが得られた．これを文脈情報として与えたものと与えないものについて平均精度を比較したところ，同義語リストを使用したものの平均精度は 40.3%，使用しないものの平均精度は 41.1%であった（同義語リストに含まれるものは評価から除外した）．同様に出現頻度が 300 以上のものについて比較したところ，同義語リストを使用したものの平均精度は 38.6%，使用しないものの平均精度は 39.4%であった．同義語リストを使用する方が精度が向上すると予想していたが，結果としては逆で若干（0.8%）精度は悪かった．原因は，定かではないが，多義語が悪影響した可能性が考えられる．次に出現頻度が 600 以上のものについても同義語リストに含まれるものは文脈語に含めて比較したところ，同義語リストを使用したものの平均精度は 41.7%，使用しないものの平均精度は 41.0%であった．この場合は，同義語リストを使用した場合の精度は若干向上したが，辞書の作成段階で同義語リストを使用する方がよいとは云えないものであった．

表 5: 同義語辞書

見出し語	登録語	
APP	Approach	進入
Approach	APP	進入
進入	APP	Approach
CRZ	巡航	
巡航	CRZ	

表 6: 同義語リスト

見出し語	変換語
APP	進入
Approach	進入
CRZ	巡航

6 結論および今後の課題

本論文では，特定分野における同義語辞書作成支援システムを提案した．対象の語句の前後に出現する語の文脈情報のみを使用したが，人間の辞書支援システムとしては，十分に機能することを実験の結果確認した．

今後の課題としては，以下が挙げられる：

- 本論文では，パイロットレポートの本文を用いて実験したが，本文よりも表題に略語が頻繁に用いられているので，同様の実験を行う．ただし，表題は体言止めになっている場合が多いので，後ろの文脈が使用できないものが多いと考えられる．
- 文脈語数を制限することにより，性能を低下させないでスペース，計算量削減できるかを実験する．
- 実用化する場合には，英語とカタカナ，英語とその略語の対については文字情報を用いて対応関係を絞み精度を向上させる．
- 多義語の処理については，クエリに対する典型的な文脈 (ベクトル) 情報が得られていれば，そのクエリが出現する文脈から多義性を解消できる可能性がある．例えば，「Noise について Cabin に問い合わせたところ，Cabin での Noise は，Door 近くからであることが判明した」という文の 1 番目のの Cabin は，客室乗務員のことであり，2 番目の Cabin は，客室のことである．

7 謝辞

専門用語自動抽出システムは，東京大学中川研究室・横浜国立大学森研究室で開発された用語抽出システムを使用させて頂きました．ここに感謝いたします．

参考文献

[1] A.S.Schwartz and M.A.Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing*, pages 8:451–462, 2003.

[2] Kevin Knight and Jonathan Graehl. Machine transliteration. *Computational Linguistics*, 24(4):599–612, 1998.

[3] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[4] Takeshi Masuyama and Hiroshi Nakagawa. Web-based aquisition of japanese katakana variants. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference*, pages 338–344, 2005.

[5] Kiyonori Ohtake and Youichi Sekiguchi. Detecting transliterated orthographic variants via two similarity metrics. In *Proceedings of Coling 2004*, pages 709–715, 2004.

[6] J. Pustejovsky, J. Castao, B. Cochran, M. Kotecki, M. Morrell, and A. Rumshisky. Extraction and disambiguation of acronym-meaning pairs in Medline, unpublished manuscript, 2001.

[7] Akira Terada, Takenobu Tokunaga, and Hozumi Tanaka. Automatic expansion of abbreviations by using context and character information. *Inf. Process. Manage.*, 40(1):31–45, 2004.

[8] 酒井浩之，増山繁．略語とその原型語との対応関係のコーパスからの自動獲得手法の改良．自然言語処理, 12(5):207–231, 2005.

[9] 阿玉泰宗，橋本泰一，徳永健伸，田中穂積．日英言語横断情報検索のための翻訳知識の獲得．情報処理学会論文誌：データベース, 45(SIG 10):37–48, 2004.

[10] 中川裕志 , 森辰則 , 湯本紘彰. 出現頻度と連接頻度に基づく専門用語抽出. 自然言語処理, 10(1):27–45, 2003.