自 然 言 語 処 理 149－3
（２００２．５．２３）

系列パターンマイニングによる対訳表現抽出

山本 薫　　工藤 拓　　坪井祐太　　松本裕治

奈良先端科学技術大学院大学 情報科学研究科

〒 630-0101 奈良県生駒市高山町 8916-5

{kaoru-ya,taku-ku,yuuta-t,matsu}@is.aist-nara.ac.jp

本稿では、対訳コーパスからの対訳表現抽出をデータマイニング分野で研究されている系列データマイニング問題とみなし適用する。統計的尺度に基づく対訳表現抽出は、対訳コーパスで共起頻度が高い候補パターンは、独立頻度も高いという直観に基づいている。従来方法では、言語的手がかりを利用して単言語コーパスで独立頻度が高い候補パターンを生成し、高頻度な単言語パターン間の共起頻度を数えあげるという二段階の処理が必要であった。本手法では、単語分割された対訳文を連結させた対訳系列から系列パターンマイニングを適用することにより、組合せ爆発を回避しながら、ある閾値以上頻出する単言語パターンの独立頻度と二言語パターンの共起頻度の効率的な数えあげを実現した。
さらに、系列パターンマイニングを解く PrefixSpan アルゴリズムに品詞による言語制約を導入した。具体的には、自立語だけの系列と機能語も考慮した系列から候補パターンとして生成し、比較実験を行なった。自立語だけの系列の場合、計算時間は短く、名詞句の対応には有用だが、その他の複数語対応には有効ではない。一方、機能語も考慮した場合、計算時間が長くなるが、訳し分けに有用な熟語パターンが多く抽出でき、計算速度と品質というトレードオフが観察された。本提案は精度向上という課題も残されているが、いままで困難とされてきた訳し分けに有用な複数語の対訳表現が抽出できた。

キーワード：対訳表現抽出、系列パターンマイニイング

Bilingual Lexicon Extraction: A Data Mining Approach

Kaoru Yamamoto　　Taku Kudo　　Yuta Tsuboi　　Yuji Matsumoto

Graduate School of Infomation Science, Nara Institute Science and Technology

8916-5 Takayama, Ikoma Nara 630-0101 Japan

{kaoru-ya,taku-ku,yuuta-t,matsu}@is.aist-nara.ac.jp

This paper proposes a sequential pattern mining to extracting bilingual lexicons from parallel corpora. Co-occurrence based bilingual extraction follows an intuition that frequent bilingual patterns in parallel copora are also frequent in each half of monolingual corpus. Previous approaches follow a two-step candidate generation and counting: generate candidate patterns independently in each monolingual corpus, and then count their co-occurrence frequency. Our approach concatenates parallel sentences into bilingual sequences from which frequent sequential patterns are mined. By applying sequential pattern mining, we avoid the combinatorial explosion in candidate generation and are able to count independent and co-occurrence frequencies simultaneously.

Keywords：bilingual lexicon extraction, sequential pattern mining

1 はじめに

1990 年代から大量の電子データが入手可能になったことにより、コーパスに基づく手法が形態素解析、構文解析などさまざまな分野で適用されている。機械翻訳も例外ではなく、統計的機械翻訳や例に基づく機械翻訳など、データに基づく機械翻訳が採用されている。これらのデータに基づく枠組では、対訳データから翻訳知識を獲得することを前提としている。しかし、辞書構築などの翻訳知識の全てを人手で獲得するのは限界があり、それらの作業の自動化が望まれている。

本稿では、データマイニングの一種である系列パターンマイニングによる対訳コーパスからの対訳表現抽出について述べる。翻訳は、イディオムなど熟語で対応しているものがあるため、単語対応のみではできない。また、分野特有の用語や言い回しなどの定型表現は複数の語から構成され、その並びは必ずしも連続とは限らない。翻訳処理では、これらの複数語表現をパターンとしてそのまま訳出したいという要望がある。武田は、語彙や定型表現とそれらの対訳をパターンとして表現し、翻訳処理でそれらのパターンに記述してある適応可能な条件を照らしあわせながら同期導出していく、パターンに基づく機械翻訳 [14] を提案した。今後、データに基づく枠組が主流になるにつれ、単語対応以上の対訳表現抽出は大切な基礎技術となる。

このような背景から、本研究の目標を、先行研究で比較的多く取り上げられている単語対応 [10] ではなく、複数語から構成される対訳表現の抽出と定める。対訳コーパス中に頻出する複数語から構成されるパターンを効率よく抽出するために、系列パターンマイニングを適用する。系列パターンマイニングは、データ中にある閾値以上存在する必ずしも連続ではない列をすべて抽出する。本稿では、(1) 順序を保持した連続及び非連続なパターンが抽出できること、(2) 高頻度のパターンのみを生成し数え上げの対象とする、といった系列パターンマイニングの特徴を活かし、頑健な複数語対応を提案する。

次節では、複数語対応 (コロケーション対応) の従来手法を検証し、本研究の特色を述べる。3 節では、系列パターンマイニングによる対訳表現抽出を提案する。4 節で、実験手法とその結果を報告し、5 節で考察を行なう。6 節でまとめと今後の展望を述べる。

2 複数語対応

複数語対応の初期の研究は、Kupeic[6] や熊野ら [5] によるもので、対象とする言語現象を名詞句に限定したものであった。その後、特定の言語現象に限定しない複数語対応を対象とする研究として、Smadja らが、英語とフランス語の二言語コロケーションの獲得方法[1]を報告した [13]。論文では、複数語対応の重要性を唱えるとともに、コロケーションを連続コロケーションと非連続コロケーションの 2 つに分類している。本研究で目指す複数語対応も、Smadja らの分類に従い、特定の言語現象に限定しない。

連続コロケーションとは、表層的に連続した複数語対応を指す。例として、固有名詞 ⟨New York, ニューヨーク⟩、have や make など light verb を含む連語 ⟨have a dinnar, 食事をする⟩、専門用語などが挙げられる。一方、非連続コロケーションとは、間に他の語や表現を含む複数語対応を指す。例えば、定型表現 ⟨as... as possible, できるだけ... ⟩ がある。複数語対応の大半は、連続コロケーションに分類され、抽出精度の向上が必須事項である。一方、非連続コロケーションは、数も少なく、抽出するのが困難である。

本研究は、統計的尺度に基づく対訳表現抽出の研究の中で、Smadja ら [13]、北村ら [17]、春野ら [4] の 3 つの先行研究を念頭に置いているので、それらについて詳しく紹介する。

Smadja らは Xtract を使って、名詞-名詞、動詞-名詞、形容詞-名詞などの英語のコロケーションを推定し、それと対応するフランス語単語を Dice 係数を基にした類似度を考慮しながら対応をとった。片方の言語でのコロケーションを固定をしないと相手言語との対応がとれないという非対称な方法である。春野らは word-level sorting で単言語のコロケーションをボトムアップに生成し、相互情報量を使って二言語間のコロケーションの抽出を行なった。この手法は、Smadja らの手法とは違い、対称であるが、あらかじめ単言語内の処理で対訳になりそうなコロケーションの候補生成を絞っている。これらの手法では、対訳コーパス中のある頻度以上出現する二言語コロケーションを網羅的に考慮していないという共通の問題がある。

一方、北村らは自立語から構成される任意長の単語列対応を抽出した。単言語コーパスに対して形態素解析を行ない、自立語のみを抽出し、連続して出現する自立語列 (自立語の N-gram) を候補として網羅的に生成する。候補集合のなかから、重み付き Dice 係数を基に対訳表現を抽出する。この手法も対称性を有しており、Smadja らや春野らの手法と違い網羅的に候補を考慮するという点で優れている。しかし、候補を自立語の N-gram として生成としているため非連続コロケーションが抽出できない。

複数語対応を議論した先行研究と本研究の特色を表 1 にまとめる。提案手法の特色は、対称性もあり、かつ、ある頻度以上出現する連続および非連続コロケーションを網羅的に生成する点である。

従来研究において、非連続コロケーションを網羅

[1]厳密には、コロケーションと複数語表現には違いがあるが、本稿では、複数語対応とコロケーション対応を同意語として使う。

手法	特色	連続	非連続	網羅性	対称性
Kupeic(1993)[6]	名詞句に限定	○	×	△	○
熊野ら (1994)[5]	辞書にない名詞句に限定	○	×	△	○
Smadja(1996)[13]	語の共起関係とその距離による統計量を使用	○	○	×	×
北村ら (1997)[17]	任意長の自立語単語列を対象	○	×	○	○
春野ら (1998)[4]	word-level sorting による候補の生成と絞り込み	○	○	×	○
本手法	系列パターンマイニングによる候補の生成と数えあげ	○	○	○	○

表 1: 複数語対応: 各手法の比較

的に考慮しない理由は 2 点あったと筆者らは考える。まず、非連続コロケーションを網羅的に生成する単純な方法がなかった。次に、仮に単言語で網羅的に連続および非連続コロケーションの候補生成をしても、組合せ爆発が起こると容易に予想できた点にある。

予備的な実験として、非連続コロケーションを考慮した場合、どのぐらい組合せ数が増加するのかを調査した。本研究で使用する対訳コーパス (英語単語数: 44743、日本語単語 (形態素) 数:186470) に対して、後述する PrefixSpan アルゴリズムを使い、単言語コーパス内で 3 回以上出現する連続コロケーションと非連続コロケーションの生成を行なった。表 2 に結果をまとめる[2]。非連続コロケーションを考慮することにより、対訳候補の組合せ数は、(35,857 × 44,343 → 1,463,744 × 7,654,621) となり、単純な方法ではこれらの組合せを考慮できない。

統計的尺度に基づく対訳コーパスからの対訳表現抽出において重要なのは、共起頻度が高い組合せである。共起頻度が高い組合せは対訳コーパスの片言語側での独立頻度も高いと予測できるため、単言語での独立頻度が高いものを別々に列挙した後、両言語での共起頻度を数えあげる必要があった。表 2 からも読みとれるように、非連続コロケーションを考慮すると、組合せ爆発が簡単におこることが窺える。

提案手法では、系列パターンマイニングを適用することにより、組合せ爆発を回避しながら、対訳コーパス中に頻出する連続及び非連続の二言語コロケーションの抽出を可能にした。

3 系列パターンマイニングによる対訳表現抽出

図 1 に提案手法の全体像を示す。従来手法との違いは、次のとおりである。

- 系列パターンマイニングで連続と非連続な対訳候補パターンを組合せ爆発を回避しながら網羅的に考慮できる。
- 対訳候補パターンの共起頻度と独立頻度を同時に数えあげることができる。

[2]表 2 の連続 + 非連続は系列パターンマイニングで抽出された系列パターンの数である。連続パターンと非連続パターンを区別して数え上げていないことに注意されたい

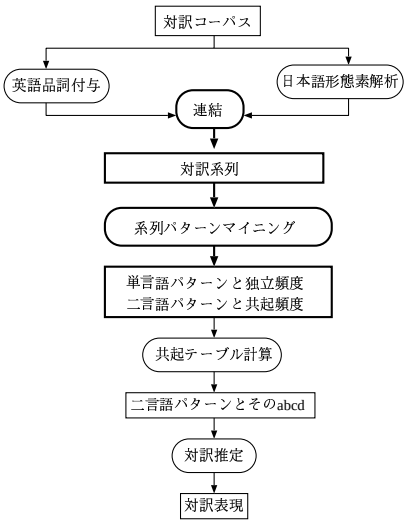

図 1: 系列パターンマイニングによる対訳表現抽出

以下では、本手法で中心となっている系列パターンマイニングについて紹介し、対訳表現抽出への応用について述べる。素直に適用すると、副作用として言語的に意味のない対訳表現が抽出される。そこで、工藤ら [15] が提案した関係関数を使った言語制約の導入について述べる。

3.1 系列パターンマイニング

系列パターンマイニングとは、データマイニングの一種で、Agrawal らによって提唱された問題である [1]。相関関係を求めるバスケット分析問題と違い、系列パターンマイニングは順序を考慮するため、テキストマイニング分野での応用が期待できる [15]。詳細な定義は文献 [1] に譲るが、系列パターンマイニングを次のように定義する[3]。

系列パターンマイニング: 系列データベース S と最低出現回数 ξ が与えられたとき、 S に ξ 以上頻出する部分系列 s の完全な集合を求める問題。

系列パターンと自然言語処理で多用される連続単

[3]Agrawal らの定義では、系列パターンマイニングは最低出現回数 ξ ではなく最小サポート Ξ による定義である。S における s のサポートとは、S 中の部分系列 s を含む系列の割合を指す。本稿では、出現回数を閾値としたいので、サポートを出現回数と読みかえて議論を進める。後で述べる PrefixSpan アルゴリズムの補題における $support_S(s)$ とは、系列データベース S に部分系列 s が出現した回数を指す。

頻度	連続のみ (英)	連続 + 非連続 (英)	連続のみ (日)	連続 + 非連続 (日)
10	1,693	29,884	2,238	215,640
9	1,952	36,239	2,562	263,780
8	2,319	45,173	3,030	332,591
7	2,801	58,411	3,663	433,259
6	3,560	79,858	4,577	594,599
5	4,733	117,447	5,986	886,479
4	7,030	284,597	8,462	1,520,614
3	11,769	812,135	13,825	3,407,659
total	35,857	1,463,744	44,343	7,654,621

表 2: 連続と非連続コロケーションの数

語列 (N-gram) との違いを表 3 にまとめる。主な相違点は、非連続コロケーションも考慮できることと、系列パターンの長さに対する制限がないことである。つまり、系列パターンは N-gram を包含する関係である。本研究の目標である複数語対応では、連続性と構成用語数という制約を取り外すことができ、好都合である。 さらに、系列パターンマイニングでは、

	連続	非連続	N の制限
N-gram	○	×	有
系列パターン	○	○	無

表 3: N-gram と系列パターンの比較

ξ 未満のパターンを生成しないため、パターンの効率的な生成と数え上げが実現できる。

良い点を強調したが、問題点もある。例えば、“My best friend wishes your father to visit” と “... and best wishes for success” からは、“best_wishes” というパターンが生成されるが、本来ならば、違うパターンとして数えあげるべきものである。本研究では、このような事例は違う日本語訳が対応しているであろうという仮定し、共起計算で類似度が低くなるため対訳抽出に悪影響を及ぼさないと判断した。

3.2 対訳表現抽出への応用

対訳文を連結させた対訳系列を集めたデータベースから系列パターンマイニングを適用すると、対訳コーパス中に ξ 以上出現する連続、非連続の候補パターンが網羅的に生成され、かつ、その独立頻度と共起頻度が効率的に数えあげられる。以上の議論を詳しく述べる。

対訳文の連結: 対訳文の英文と和文をそれぞれ単語に分割し、それらを連結させ対訳系列を作成する。候補生成のために対訳文を連結させるというのは、従来方法にはみられない発想である。この連結操作により単言語パターンと二言語パターンが同時に生成され、かつ、それらの独立頻度と共起頻度の効率的な数えあげが可能になる。

系列パターンマイニング: 連結された対訳系列から系列パターンマイニングにより候補パターンの生成と数えあげを行なう。図 2 に示すように、単言語パターン (E,J) も二言語パターン (EJ) も同時に生成される。さらに、単言語パターンと二言語パター

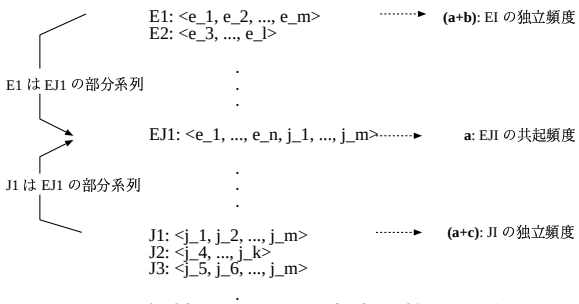

図 2: 候補パターンの生成と数えあげ

ンの独立出現回数と共起出現回数が効率的に数えあげられる。従来手法では、単言語パターンの独立出現回数を数えあげた上で頻度で足切りをし、高頻度の単言語パターンの組合せの共起出現回数を数えあげると必要があったが、本手法では、独立出現回数と共起出現回数を同時に数えあげられる。

共起テーブルの計算: 表 4 の共起テーブルで解釈すると太文字の部分が直接的に数えあげられ、対訳文の全体数がわかっているので、他の値 (b, c, d) も即時に計算可能である。共起テーブルを使えば、松本ら [9] が報告しているように様々な類似度が使用可能である。

	J が**有**	J が**無**	
E が**有**	**a**	b	**a + b**
E が**無**	c	d	
	a + c		

表 4: 共起テーブル

組合せ爆発を回避しながら網羅的な非連続コロケーションの生成をする対訳表現抽出が可能になり、先行研究の問題点を解決できたと考える。

3.3 言語制約の導入

系列パターンマイニングによって網羅的に非連続コロケーションが生成可能になったが、副作用として

言語的にあまり意味のない対訳候補パターンが生成される現象がある。この問題に対処するために、筆者らは、系列パターンの生成に言語制約を導入した。

言語制約導入の実現は、系列パターンの生成アルゴリズムに依存する。我々は、系列パターンマイニングを効率的に解く、さらに、言語制約を導入しやすいという理由から PrefixSpan[11] を採用した。更に、工藤ら [15] が提案した関係関数を、言語制約を定義する関数とした。以下では、PrefixSpan を簡単に紹介し、本研究で使った具体的な言語制約について述べる。

3.3.1 PrefixSpan

PrefixSpan は、系列パターンマイニングを効率的に解く方法として、Pei らによって考案された [11]。これは、頻出する *prefix* で系列データベースを分割し深さ優先にパターンを発見する手法であり、Agrawal らが考案した generate-and-test に基づく方法 [1] より、効率よく解けることが報告されている。

PrefixSpan の理解に不可欠な定義と補題を引用する[4]。

定義 α を S に存在する系列パターンとする。α-projected データベース ($S|_\alpha$) とは S に存在する系列のうち、α を *prefix* とする系列 s の *postfix* の集合である。

補題 系列パターン α と β が系列データベース S の中にあるとする。さらに、α は β の *prefix* で、b をアイテムとするとき、b が α に後続している、$\beta = \alpha b$、であるとする。このとき、次の関係が成り立つ。

1. $S|_\beta = (S|_\alpha)|_b$
2. $support_S(\beta) = support_{S|_\alpha}(b)$

続いて、PrefixSpan アルゴリズムを紹介する。
Subroutine *PrefixSpan*(α, $S|_\alpha$)

- パラメータ:
  α: 系列パターン, $S|_\alpha$: α を *prefix* にもつ投影データベース
- 方法:
  1. **系列パターン α から 要素 b へ投影可能**で、かつ、$support_{S|_\alpha}(b) \geq \xi$ を満たす b の集合 B を見つける。
  2. B が空集合 ϕ なら終了。
  3. B の各要素 b について
     (a) 拡張系列パターン αb とその頻度 $support_{S|_\alpha}(b)$ を出力する。
     (b) 拡張系列パターン αb を *prefix* にもつ投影データベース $(S|_\alpha)|_b$ を作り、*PrefixSpan*(αb, $(S|_\alpha)|_b$) を再帰的に呼ぶ。

[4] prefix,postfix などの詳細定義は、オリジナル論文 [11] を参照されたい

上記の PrefixSpan アルゴリズムで、言語制約は、系列パターン α から要素 b へ投影が言語制約を満足しているか否かで定義する。制約を満たしている場合のみ、数えあげの対象となる。本研究では、要素は単語とおき、単語の言語制約を品詞分類 (自立語と機能語の分類) に基づいて定義した。

3.3.2 自立語と機能語

日本語のサ変名詞が英語では動詞に訳されるなど、翻訳において品詞が対応するとは限らない。それより、各言語において、自立語と機能語に分類した場合、自立語は一語一句翻訳されやすいという性質がある。前出の北村らは、この仮定に基づいて、自立語のみから構成される任意長の連続 N-gram を候補パターンとして生成している。本研究では、自立語のみの候補パターンと機能語を含む候補パターンの 2 つの関係関数を定義し、それぞれの結果を比較した。

表 5 に、本研究で用いた品詞による自立語と機能語の分類を示す。英語の品詞体系は PennTreebank[12] に、日本語の品詞体系は茶筌で利用している第一階層 [7] に基づく。

	自立語	機能語
英	NN,NNS,NNP,NNPS	PRP,PRP$,**POS**,DT
	VB,VBP,VBN,VNZ,VBG	IN,TO,**MD**,CC,CD
	JJ,JJR,JJS	WRB,WDT,WP,RP
	RB,RPR,RBS	UH,FW,PDT,EX
		.,” “:
日	名詞, 動詞, 副詞, 形容詞	助詞, **助動詞**, 接頭詞
	接続詞, 未知語, 感動詞	**連体詞**, フィラー, 記号

表 5: 自立語と機能語の分類

3.3.3 自立語のみの候補パターン生成

関係関数で定義すべきことは、パターンから要素への射影が可能か不可かという関数である。言い換えると、パターンの最後の要素から要素を追加してパターンを拡張できるかどうかを定義すればよい。自立語のみの候補パターンは、その単語の品詞が自立語であれば、数えあげの対象になるので射影が許可される。以下では、3.3.1 節において、パターンに追加する要素 b に対して、b が自立語なら投影可能という制約を課した。

3.3.4 機能語も考慮した候補パターン生成

名詞句以外の定型表現を考える場合、機能語も考慮する必要がある。しかし、単純に適用すると、連接していない機能語のみから構成される候補パターンなど、対訳表現としてはふさわしくないものも多く抽出されてしまう。

そこで、表 6 に同一言語のときに適用する連接制約を導入した。表 6 では、パターンの末尾の要素 i と

連接表 (英)	自立語	POS を除く機能語	POS
自立語	○	○	連接 ‡
MD を除く機能語	○	連接 †‡	連接 †‡
MD	連接 †	連接 †‡	連接 †‡

連接表 (日)	自立語	助動詞を除く機能語	助動詞
自立語	○	○	連接 ‡
連体詞を除く機能語	○	連接 †‡	連接 †‡
連体詞	連接 †	連接 †‡	連接 †‡

表 6: 機能語を考慮するときの言語制約 (上:英語, 下:日本語)

パターンに追加する要素 j が投影可能かどうかを定義している。i の品詞条件は縦方向に、j の品詞条件は横方向に記す。

一般則として、同一言語でパターンの末尾の要素 i と追加する要素 j が両方とも機能語の場合、連接していれば投影可能 (表 6 の †‡)、そうでなければ投影不可という言語制約を課した。

さらに、日本語の助動詞は直前の自立語 (動詞) と同時に考慮すべきであり、連体詞は直後の自立語 (名詞) と同時に考慮すべきである。同様に、英語の POS('s など所有を表す) は直前の自立語 (名詞) と、MD(will などの助動詞) は直前の自立語 (動詞) と同時に考慮すべきである。 これを反映するために、機能語を細かく分類した。日本語の場合、パターンの末尾の要素 i が自立語で、パターンに追加する要素 j が助動詞 (機能語) の場合、i と j が連接している場合のみ投影可能 (表 6 の ‡) とした。また、パターンの末尾の要素 i が連体詞 (機能語) で、パターンに追加する要素 j が自立語の場合、i と j が連接している場合のみ投影可能 (表 6 の †) とした。英語の場合、パターンの末尾の要素 i が自立語で、パターンに追加する要素 j が POS(機能語) の場合、i と j が連接している場合のみ投影可能 (表 6 の ‡) とした。また、パターンの末尾の要素 i が MD(機能語) で、パターンに追加する要素 j が自立語の場合、i と j が連接している場合のみ投影可能 (表 6 の †) とした。

4 実験と結果

実験では、日経ビジネスレター集 9269 文を利用した [16]。英語は TnT [3] で品詞タグつけを行ない、日本語は茶筌 [8] で形態素解析を行ない、対訳系列を準備した。表 7 に実験で利用したパラメータとその結果をまとめる。

表 7 にある PrefixSpan を実行する際のパラメータの大半は実験的に選定した。最低出現回数とは、系列パターンが対訳系列データベース中に出現した頻度に対する閾値である。今回の実験では、3 に設定した。一方、最短長とは、系列パターンの長さに対する閾値である。今回の実験では、複数語の対応が主な目的なので、英語 2 単語かつ日本語 2 形態素以上のものを対象とした[5]。

	自立語のみ	機能語を含む
系列の平均長	32.51	32.51
内訳 (英文)	14.27	14.27
内訳 (和文)	18.23	18.23
最低出現回数	3	3
最短長	英 2 和 2	英 2 和 2
対訳パターン候補数	96,868	59,204,224
英語パターン候補数	14,822	201,952
日本語パターン候補数	64,187	642,760
計算時間	13m05s	22h47m46s
内訳 (PrefixSpan)	12m22s	11h26m43s
内訳 (共起計算)	0m12s	2h21m48
内訳 (抽出)	0m31s	8h59m15
精度	55.93%	34.96%(55.89%)
抽出数	295	2145(492)
正解数	165	750(275)
内訳 (連続)	132	705(235)
内訳 (非連続)	33	45(40)

表 7: 実験で利用したパラメータと結果

対訳抽出は、PrefixSpan、共起計算、抽出という 3 つのモジュールで構成される。モジュールはすべて C++で実装し、Pentium III(1266MHz) メモリ 1GB の Linux マシンで実行した。共起計算とは、共起テーブル 4 に基づき、重みつき Dice 係数を類似度とした。

$$\log_2 a \ \times \ \left(\frac{2a}{(a+b)+(a+c)} \right)$$

抽出モジュールは、共起頻度 a に対する閾値を 50, 40, 30, 20, 10, 9, 8, 7, 6, 5, 4, 3 と下げてゆき、各閾値で類似度が $\log_2 a$ 以上でかつ、英語と日本語の両方が別の候補とより高い類似度で対訳表現として抽出されていない場合、新たに対訳表現を抽出するという簡単な実装にした。

精度は、1 名の日英のバイリンガルスピーカーが判断した結果を元に算出した。機能語を含む列の () は、最低出現回数を 4 に設定したときの結果を載せる。

[5]複数語対応なので、英語 2 単語または日本語 2 形態素以上のものを対象にするのが素直であるが、考慮する候補数が多くなりすぎるため、英語 2 単語かつ日本語 2 形態素以上のものを対象とした。

表 8 に自立語のみを数えあげの対象とした候補パターンの生成から抽出した正解の複数語対応例を示す。表 9 に候補パターンの生成に機能語も考慮した場合の正解の複数語対応例を示す。

5 考察

今回の実験では、困難とされてきた複数語対応のみを対象としたため、精度という点ではあまりよくない[6]。主な原因は、系列パターンマイニングによる候補生成ではなく、対訳推定 (抽出モジュール) にあると考える。特に、同じ対訳文から生成された候補パターン群の類似度が同じになり、それらの対応が決定不能になる場合が多い。この傾向は、系列で考慮する要素数が多くなる機能語を含む場合に顕著に現れる。不正解の大半は、これが原因であり、よい例は半正解に近いが、悪い例はよく共起しているが対訳としては認められないのもある。精度向上には、類似度の定義を含めた包括的な対訳同定モジュールの改良が必要である。翻訳支援の立場では、人間チェックを前提としているため、対訳抽出の重点は、精度より被覆率にある。このような場合、網羅的に複数語対応を実現する提案手法は有効だと考える。

PrefixSpan アルゴリズムは、大規模なデータベースからの系列パターンを抽出する目的で設計されたため、提案手法もデータの増加に堪えうると考える。筆者らは、手元にある一万文程度の対訳コーパスしか実験できなかったが、提案手法は、大量の対訳コーパスにも適用可能であると推測する。また、比較的浅い自然言語処理 (単語区切りと品詞付与) でも実現できることも勘案すると、提案手法は、頑健な手法であると言える。

次に、言語制約の有効性を検討する。表 7 の計算時間をみると自立語のみの系列パターンマイニングの方が、高速に対訳表現が抽出される。これは、考慮する候補パターンが格段に少ないため、計算量を押えることができた。機能語を含めると表 6 の言語制約を入れても、数多くの候補パターンが生成され、計算時間がかかる。

抽出された対訳表現の品質では、表 8 と表 9 より、機能語を考慮した方が対訳表現として興味深いものが抽出できた。自立語のみの候補パターン生成で抽出されたものは、名詞表現が多く、句境界を意識せずに対訳表現を抽出するため、長めだが言語的はあまり意味のない自立語列が抽出された。一方、機能語を考慮した場合は、名詞表現以外にも、ビジネス英語教材の熟語パターンに登録されそうな典型的な定型表現が抽出できた。⟨for your information, ご参考 ⟩ や ⟨on behalf of, を 代表 し ⟩ など、機能語を含む連続コロケーションの定型表現が抽出されている。さらに、⟨impressed with, に 感銘 ⟩ や ⟨apologize for, こと お詫び ⟩ など、連続コロケーションではあるが、動詞とその共起する前置詞の組が抽出できた。動詞の格フレームほど強力ではないが、パターンを翻訳処理に適用するときの訳しわけの手がかりになるのではと考える。さらに、少数ではあるが、非連続コロケーションも抽出されている。例えば、⟨please ... at earliest convenience, ご 都合 つき 次第 ... を 下さい ⟩ は、該当する対訳文では依頼する内容が異なっていた。また、⟨be staying at ... hotel, ... ホテル に 滞在 ⟩ は、滞在するホテル名が異なり、抽象化の効果を得た。

まだまだ課題が多い提案手法ではあるが、今後は、品詞などの素性を含めた対訳表現抽出を目指したい。例えば、⟨every effort to **VB**, **動詞** よう あらゆる 努力 ⟩ という対訳パターンが抽出されるように拡張したいと考える。また、データマイニング分野では、半構造化データから頻出する木やグラフを発掘するという手法も提案されている [2]。対訳表現は、部分木で表現した方が自然なので、これらの手法が適用可能であるか検討してゆきたいと考える。

6 まとめ

本稿では、系列パターンマイニングによる対訳表現抽出を報告した。単語分割された対訳文を連結させた対訳系列から系列パターンマイニングを適用することにより、組合せ爆発を回避しながら、ある閾値以上頻出する単言語パターンの独立頻度と二言語パターンの共起頻度の同時数えあげを効率的に実現した。

さらに、系列パターンマイニングの効率的なアルゴリズムである PrefixSpan に、品詞による言語制約を導入した。具体的には、自立語だけの系列と機能語も考慮した系列から候補パターンとして生成し、比較実験を行なった。自立語だけの系列の場合、計算時間は短いが、名詞句以外の複数語対応には有効ではない。一方、機能語も考慮した場合、計算時間が長くなるが、訳し分けに有用な熟語パターンが多く抽出でき、計算時間と品質というトレードオフが観察された。精度向上の余地があるが、機能語を含む系列パターンマイニングでは、いままで困難とされてきた訳し分けに有用と思われる熟語パターンの対訳表現が抽出できた。

謝辞日経ビジネスレター例文集の研究利用許諾をいただいた日本経済新聞社に感謝の意を表する。また、ツールを公開していただいている Thostem Brants 氏と茶筌開発集団に感謝する。

参考文献

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 1995 International Conference of Very Large DataBases (VLDB'95)*, pp. 3–14, 1995.

[6] 単語対応での実験 (PrefixSpan の最大長パラメータを英 2 和 2) では、抽出数 961 の内、896 が正解で 93% の精度であった。同じデータを北村らの手法で試したところ、636/662 (96%) であり、提案手法の方が精度は下がるが正解の絶対数が向上した。

	英語	日本語
連続	look forward	楽しみ
連続	foreign exchange	外国 為替
連続	enclosed envelope	同封 封筒
連続	annual report	年次 報告
非連続	wish ... success	成功 お祈り
非連続	please ... soon	早く ... 下さい
非連続	thank ... letter	手紙 ... ありがとう

表 8: 自立語のみの候補パターン生成で抽出された正解例

	英語	日本語
連続	impressed with	に 感銘
連続	apologize for	こと お詫び
連続	end of this month	今月 末
連続	your fax of	付け ファクス
連続	near future	近い 将来
連続	for your information	ご 参考
連続	look forward to	を 楽しみ
連続	thank you letter of	の 手紙 ありがとう
連続	on behalf of	を 代表 し
連続	distressed to	心 痛め おり ます
連続	every effort to	よう あらゆる 努力
非連続	please ... at earliest convenience	ご 都合 つき次第 ... を 下さい
非連続	i ... be in japan to	日本 に 滞在 ...
非連続	be staying at ... hotel	... ホテル に 滞在

表 9: 機能語も考慮した候補パターン生成から抽出された正解例

[2] T. Asai, K. Abe, S. Kawasoe, H. Sakamoto, and S. Arikawa. Efficient substructure discovery from large semi-structured data. Technical report, Department of Informatics, Kyushu University, 2001.

[3] T. Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference*, 2000.

[4] M. Haruno, S. Ikehara, and T. Yamazaki. Learning bilingual collocations by word-level sorting. In *ACL/COLING-96*, pp. 525–530, 1996.

[5] A. Kumano and H. Hirakawa. Building an mt dictionary from parallel texts based on linguistic and statistical information. In *31st Annual Meeting of the Association for Computational Linguistics*, pp. 76–81, 1994.

[6] J. Kupiec. An algorithm for finding noun phrase correspondences in bilingual corpora. In *31st Annual Meeting of the Association for Computational Linguistics*, pp. 23–30, 1993.

[7] Y. Matsumoto and M. Asahara. Ipadic users manual. Technical report, Nara Institute of Science and Technology, 2001.

[8] Y. Matsumoto, A. Kitamuchi, T. Yamashita, H. Matsuda, K. Takaoka, and M. Asahara. Morphological analysis system chasen version 2.2.1 manual. Technical report, Nara Institute of Science and Technology, 2000.

[9] Y. Matsumoto and T. Utsuro. Lexical knowledge acquisition. In *Handbook of Natural Language Processing*, pp. 563–610. Marcel Dekker, 2000.

[10] I.D. Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001.

[11] J. Pei, B. Han, J. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hau. Prefixspan: Mining sequential patterns efficiently by prefix- projected pattern growth. In *Proc. of International Conference of Data Engineering (ICDE20 01)*, pp. 215–224, 2001.

[12] B. Santorini. Part-of-speech tagging guidelines for the penn treebank project. Technical report, LDC, 1991.

[13] F. Smadja, K.R. McKeown, and V. Hatzuvassiloglou. Translating collocations for billingual lexicons: A statistical approach. In *Computational Linguistics*, Vol. 22(1), pp. 1–38, 1996.

[14] K. Takeda. Pattern-based context-free grammars for machine translaction. In *34th Annual Meeting of the Association for Computational Linguistics*, pp. 144–151, 1996.

[15] 工藤拓, 山本薫, 坪井祐太, 松本裕治. 言語情報を利用したテキストマイニング. In *NL-148*, pp. X–Y, 2002.

[16] 田久保浩平, 橋本光憲. 英文ビジネスレター文例大辞典. 日本経済新聞社, 1995.

[17] 北村美穂子, 松本裕治. 対訳コーパスを利用した対訳表現の自動抽出. 情報処理学会論文誌, pp. 727–736, 1997.