情報学基礎 64－3
自然言語処理 145－3
（2001. 9. 10）

対訳表現抽出における翻訳単位の比較

山本 薫　　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,matsu}@is.aist-nara.ac.jp

本稿では、対訳コーパスから自動的に対訳表現を抽出する手法の比較を行なう。翻訳単位として生成する単位を、従来のN-gram、チャンク区切り内で閉じたN-gram、単語の係り関係を使ったN-gramの3つを用意し、その精度と被覆率を比較する。どのモデルが一番高い精度と被覆率を出すかを明らかにすると同時に、対訳表現抽出においてどんな言語情報有用であるかを探る。結果、対訳表現抽出において、チャンク区切りと単語の依存関係は、それぞれ有用で、相補しあう関係にある言語情報であることを示した。

キーワード：対訳表現抽出、対訳コーパス、N-gramモデル

A Comparative Study on Translation Units for Bilingual Lexicon Extraction

Kaoru Yamamoto　　Yuji Matsumoto

Graduate School of Infomation Science, Nara Institute Science and Technology

8916-5 Takayama, Ikoma Nara 630-0101 Japan

{kaoru-ya,matsu}@is.aist-nara.ac.jp

This paper compares various translation units for extracting translation patterns automatically from parallel corpora. Three models of translation units, namely an ordinary N-gram (baseline model), a chunk N-gram, and a dependency N-gram, are proposed and evaluated in terms of accuracy and coverage. The purpose of this research is to identify the language clues useful for the task of bilingual lexicon extraction. Our experiment shows that both chunk boundaries and dependency relations between words are effective language clues that may complement each other.

Keywords : acquisition of bilingual lexicon, parallel corpora, n-gram model

1 はじめに

統計ベース機械翻訳や用例ベース機械翻訳の開発では、対訳コーパスの有効利用が必須である。電子化された対訳コーパスは利用可能になりつつあるが、単言語コーパスと比べるとまだまだ稀な資源である。そのため、対訳コーパスから翻訳知識を最大限ひきだすことは有意義なことである。

本稿では、対訳コーパスから自動的に対訳表現を抽出する手法の比較を行なう。我々の手法は、形態素解析、チャンカー、係受け解析器など、近年の自然言語処理ツールの発展に支えられている。これらのツールは統計的手法もしくは機械学習の手法を使ってデータから直接モデルを学習したものである。これらのツールから得られる「手がかり」は誤っている可能性もあるが、翻訳単位を生成するにあたり部分的に有用な言語情報がたくさん含まれている。本手法は、これらの部分的に信頼できると思われる言語情報を活用して対訳表現を抽出することを試みる。

本稿では、翻訳単位として生成する単位を、従来のN-gram、チャンク区切り内で閉じたN-gram、単語の係り関係を使ったN-gramの3つを用意し、その精度と被覆率を比較する。どのモデルが一番高い精度と被覆率を出すかを明らかにすると同時に、対訳表現抽出においてどんな言語情報有用であるかを探る。

次節では、本稿で比較する3つの翻訳単位モデルについて述べる。3節では、対訳表現抽出アルゴリズムについて、説明する。4節と5節で、実験結果を提示し、各モデルの特徴と限界について考察する。6節でまとめる。

2 翻訳単位モデル

対訳表現を自動的に抽出するとは、一般に、ある翻訳の単位を仮定しその対応を自動的に推定するものである。本稿の議論の焦点は、どのような翻訳単位を仮定すれば、どの程度の精度と被覆率が期待できるのかという関係を明らかにすることである。

先行研究の多くは、単語を単位として対訳抽出の議論を進めてきており、最近では Melamed が単語の one-to-one assumption は、制限的な仮定ではないと主張している [6]。しかし、わかち書きされていない言語は、複合語をどう区切るかなど問題があり、わかち書きそのものに曖昧性がつきまとう。したがって、1対1対応を仮定するのは危険と判断し、対訳文から翻訳単位を生成するとき、重なりを許すことでわかち書きの揺れを吸収する方針をとる。そして、単語も連語の対応も区別なく抽出することを試みる。

先行研究で、連語対応を議論したものは、次のようなものがある。Kupiec は NP recognizer を利用して翻訳単位を名詞句に限って対応をとった [4]。Smadja らは、XTRACT system を使って英語の collocation を推定し、それと対応するフランス語の collocation の対応をとった [8]。更に、Kitamura らは任意長の単語対応を抽出し [3]、Haruno らは word-level sorting による collocation の抽出を行なった [2]。

本稿では、従来の N-gram モデル、チャンク区切り内で閉じた N-gram モデル、単語の係受け関係を使った N-gram モデルを比較し、これらのモデルから生成される翻訳単位からどのような対訳表現抽出できるか比較検討する。本手法の対訳表現抽出は 2 段階から構成される。まず、対訳コーパスから各言語ごとに N-gram の翻訳単位を生成する。そして、これらの翻訳単位の候補集合から対応する対訳表現を抽出する。本手法の特徴は、翻訳単位生成のときに重なりを許すことによりわかち書きによる曖昧性を吸収する。さらに、対訳表現は貪欲的に抽出し、処理の途中で、一旦、対訳表現と推定されたら、それと重なりあう翻訳単位候補は削除する。これにより、組合せ爆発を回避する。

従来の N-gram モデル、チャンク区切り内で閉じた N-gram モデルにおける翻訳単位の生成は、自立語のみを使って翻訳単位 (N-gram) を生成する。前置詞などの機能語は、単体では、対応するものがないため、対訳抽出ではノイズになりやすい。そのため機能語はあらかじめ範囲外とする。

機能語の定義はさまざまだが、本稿では、次の条件が満たせば機能語とみなす。(英語は Penn Treebank part-of-speech tag set [7] を、日本語は形態素解析「茶筌」の定義 [5] を採用している。)

品詞 (日本語) “名詞-代名詞”, “名詞-数”, “名詞-非自立”, “名詞-特殊”, “名詞-接尾-助動詞語幹”, “名詞-接尾-副詞可能”, “名詞-接尾-助動詞”, “接頭詞”, “動詞-接尾”, “動詞-非自立”, “助詞”, “助動詞”, “形容詞-非自立”, “形容詞-接尾”, “記号”

品詞 (英語) “CC”, “CD”, “DT”, “EX”, “FW”, “IN”, “LS”, “MD”, “PDT”, “PR”, “PRS”, “TO”, “WDT”, “WD”, “WP”

be 動詞の原形 (英語)

記号類 句読点、括弧

単語の係受け関係を使った N-gram モデルは、自立語と機能語を混合させて翻訳単位 (N-gram) を生成する。これは、単語の依存関係を考えるとき、機能語を含ませるほうが自然という判断からである。ただし、機能語のみから構成される翻訳単位など、明らかに対応関係がつかない候補は、あらかじめルールを書いて削除した。

以下に、本稿で議論する 3 つの翻訳単位について図 1 を参照しながら説明する。

Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov. 29 .

[Pierre Vinken] , [61 years] [old] , [will join] [the board] [as] [a nonexecutive director] [Nov. 29] .

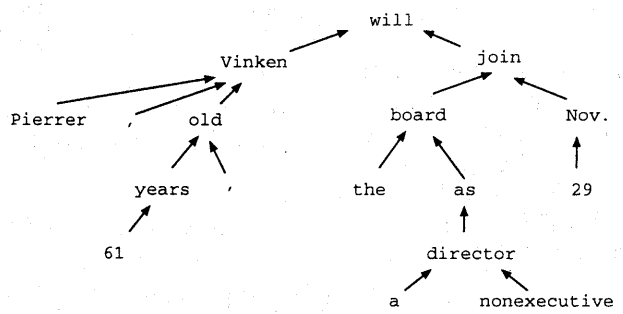

図 1: 例文: 形態素解析 (上)、チャンキング (中)、単語依存解析 (下)

Pierre	Vinken
Pierre-Vinken	Vinken-years
Pierre-Vinken-years	Vinken-years-old
Pierre-Vinken-years-old	Vinken-years-old-join
Pierre-Vinken-years-old-join	Vinken-years-old-join-board
years	old
years-old	old-join
years-old-join	old-join-board
years-old-join-board	old-join-board-nonexecutive
years-old-join-board-nonexecutive	old-join-board-nonexecutive-director
join	board
join-board	board-nonexecutive
join-board-nonexecutive	board-nonexecutive-director
join-board-nonexecutive-director	board-nonexecutive-director-Nov
join-board-nonexecutive-director-Nov.	
nonexecutive	director
nonexecutive-director	director-Nov
nonexecutive-director-Nov	Nov

図 2: 従来 N-gram モデルで生成される翻訳単位

Pierre	Vinken
Pierre-Vinken	
years	old
join	board
nonexecutive	director
nonexecutive-director	Nov

図 3: チャンク N-gram モデルで生成される翻訳単位

従来 N-gram

先行研究では、従来N-gramを翻訳単位としてみなして対訳表現を抽出するものがある [3]。このモデルは利用する言語情報が一番少なく、本稿では、ベースラインモデルとする。N の上限を固定し、uni-gram ,..., N-gram までを翻訳単位とする。実験では N の長さを 5 に設定した。図 1 から従来 N-gram モデルを使って生成される翻訳単位を図 2 に表す。

チャンク N-gram

このモデルでは、チャンク区切り情報を仮定し、区切りを越えない uni-gram,..., N-gram を翻訳単位とする。チャンクの定義は、英語の場合、chunking shared task[1] の定義を参考にし、日本語の場合、京大コーパス [10] 文節区切りに準拠したものを単位とした。

従来 N-gram とは違い、チャンク N-gram はチャンク境界を越えない。このため、N は、チャンクの長さに依存する[1]。 図 1 からチャンク N-gram モデルを使って生成される翻訳単位を図 3 に表す。

係受け N-gram

このモデルでは、文を単語依存木に解析し、枝の構成ノード (形態素/単語に相当) の範囲で uni-gram, ..., N-gram を翻訳単位とする。一般に、構文解析の曖昧性から唯一の単語依存木を決定するのは困難であるが、今回の実験では、統計的に最良な解析結果のみを採用した。

係受け N-gram モデルと先行研究の違いとして、次の 2 つをあげる。Yamamoto[9] らにおいても、係受け関係をつかって対訳表現を抽出している。しかし、この手法は、機能語を含めた文節相当にあたるもの全体を翻訳単位とみなしたため、翻訳単位が大きすぎて、データの過疎化に十分対応できなかった。抽出された対訳表現の精度は高いものの、被覆率が低いという課題が残っていた。それに比べ、係受け N-gram モデルでは、細かい形態素を単位としているので上記のようなデータ過疎性に対応できるものと考えられる。また、文中に離れたところに係受け関係がある場合には、離散的な N-gram も生成される。これらの非連続な N-gram は、従来 N-gram では生成することができない。

[1]実験で使用した対訳コーパスでのチャンクの平均長は、英語は 2.1 単語、日本語は 3.4 形態素であった。

Pierre	years
Pierre-Vinken	years-old
Pierre-Vinken-will	years-old-Vinken
Vinken	years-old-Vinken-will
Vinken-will	old
will	old-Vinken
	old-Vinken-will
the-board	a-director
the-board-join	a-director-as
the-board-join-will	a-director-as-join
board	a-director-as-join-will
board-join	director
board-join-will	director-as
join	director-as-join
join-will	director-as-join-will
	as-join
	as-join-will

nonexecutive	Nov.
nonexecutive-director	Nov.-join
nonexecutive-director-as	Nov.-join-will
nonexecutive-director-as-join	
nonexecutive-director-as-join-will	

図 4: 係受け N-gram モデルで生成される翻訳単位

3 対訳表現抽出

対訳表現抽出アルゴリズムは [9] を利用し、初めに高く設定した閾値を段階的に下げていきながら、信頼度の高い対訳表現から貪欲的に抽出する。そのため、処理途中で (閾値が高い時に) 一度対訳表現としてみなされた対訳表現は、処理のあとのほうで覆されることがない。さらに、抽出された対訳表現と重なりある翻訳単位は、そのつど、翻訳単位候補から除外されていく。

翻訳単位 p_e と p_j の類似度は重みつき Dice 係数を使って計算する。

$$sim(p_e, p_j) = (log_2 f_{ej}) \frac{2f_{ej}}{f_e + f_j}$$

f_j と f_e は、日本語側の対訳コーパスの p_e の単独出現頻度と、英語側の対訳コーパスの p_e の単独出現頻度をそれぞれ表す。 f_{ej} は、対訳コーパスにおける p_e と p_j の同時出現頻度を表す。

以下の処理を、現在の閾値 f_{curr} があらかじめ設定した最小閾値 f_{min} に到達するまで繰り返す。

1. f_{curr} 回以上 出現した英語の翻訳単位 p_e と日本語の翻訳単位 p_j のそれぞれに対して、もっとも類似度の高い対訳表現を見つける
   - 英語の翻訳単位 p_e に対して、その対訳候補集合 $\mathrm{sim}(p_e,p_{jk}) > log_2 f_{curr}$ for all k を満たす $\mathrm{PJ} = \{ p_{j1}, p_{j2}, ..., p_{jn} \}$ を求める。同様に、日本語の翻訳単位 p_j に対して、その対訳候補集合 PE を求める。

- 以下の条件が満たされれば、(p_e,p_j) を対訳表現として登録する。

$$p_j = \underset{p_{jk}\in PJ}{\text{argmax}}\ sim(p_e,p_{jk})$$
$$p_e = \underset{p_{ek}\in PE}{\text{argmax}}\ sim(p_j,p_{ek})$$

  PJ の中で p_j が一番高く、PE の中で p_e が一番高ければ (p_e,p_j) が対訳表現として登録される。

2. (p_e,p_j) と重なりあう翻訳単位について、(p_e,p_j) の同時出現している部分だけ削除する。
3. 対訳表現が抽出されないようであれば、f_{curr} を下げる。

4 実験と結果

実験では、日経ビジネスレター集5000文を利用した [11]。その内、4000文で対訳表現を抽出し、正解は人手で評価し、精度を計算した。残りの1000文で、どの程度正解対訳ペアで置き換えるかを測定し、被覆率を計算した。

対訳コーパスは、文対応していること以外情報がないので、自然言語処理ツール (形態素解析、チャンキング、係受け解析) を使って、単語のわかち書き、文節区切り、統語的な依存構造などの言語情報を推定した。これらのツールのほとんどは、統計手法、もしくは、機械学習に基づいてデータから直接モデルを学習しており、約88%の解析精度があるものを利用した[2]。

コーパス中に2回以上出現した翻訳単位のみを上述の対訳抽出アルゴリズムの対訳候補集合の要素とみなす。つまり、本手法では、対訳コーパス中に同時に出現した回数が一回きりの対訳表現は抽出されない。アルゴリズムの統計的な側面を考えるとこの代償は無視できるものと考える。表1に、各モデルで生成された翻訳単位の異なり数を示す。従来N-gramとチャンクN-gramは自立語のみで翻訳単位を生成したが、係り受けN-gramは機能語も混合させて翻訳単位を生成したため、異なりに差がある。チャンクN-gramは、チャンク区切りの長さに影響されるため、Nが固定長である従来N-gramより異なり数が少ない。

model	英語	日本語
従来 N-gram	4286	5817
チャンク N-gram	2942	3526
係受け N-gram	15888	10229

表 1: 翻訳単位の異なり数

[2] 日本語は、茶筌、yamcha, Cabocha を使い、英語は、MX-POST, Collins parser を使った

以下に対訳表現抽出における閾値の制御を示す。3節で述べたように、閾値 f_{curr} の初期値は100に設定し、段階的に下げていき、2 (f_{min}) になるまで繰り返し、類似度 $\mathrm{sim}(p_e,p_j)=1$ で止めた。

$$f_{curr} = \begin{cases} f_{curr}/2 & (f_{curr} > 20) \\ 10 & (20 \geq f_{curr} > 10) \\ f_{curr}-1 & (10 \geq f_{curr} > 2) \end{cases}$$

実験の結果は、精度と被覆率の2つの尺度で評価する。精度とは、対訳抽出アルゴリズムで抽出された対訳表現の中で、人間が正しいと判断した対訳表現の占める割合である。これは、異なりで計算している。一方、被覆率は、未知のテストデータに対して、抽出された正解対訳表現の適用度を示す。テストデータの形態素数から抽出された正解対訳表現で置き換え可能な形態素数の割合をのべ数で計算している。これらの精度と被覆率という尺度は、それぞれ、Melamed の precision と percent correct に対応している。精度は、4000文の学習データから抽出された対訳表現に対して人手で判断しており、被覆率は、1000文のテストデータに対して自動的に計算した。

表2、表3、表4に、各モデルの精度を閾値ごとに示す。"f_{curr}" は、閾値 (アルゴリズムのステップ) を示す。"e" は、閾値が "f_{curr}" のときに抽出された対訳表現の数、"c" は閾値が "f_{curr}" のときに正解と認定された対訳表現の数を表す。対訳表現の正解は、一人のバイリンガルスピーカーに判定してもらった。"acc" は精度を示す。'e"、"c"、 "acc" の累計結果は'で示す。

表5、表6、表7に各モデルの被覆率を閾値ごとに示す。前述のように "f_{curr}" は、閾値 (アルゴリズムのステップ) を示す。括弧は言語を表し、"E" は英語で"J" は日本語である。"found" は正解対訳表現に置き換え可能な対訳表現に含まれる単語総数を指す。"ideal" は、対訳表現抽出アルゴリズムで抽出される可能性のある単語の上限を示す。これは、もとの対訳コーパスで少なくとも "f_{curr}" 回以上同時出現した翻訳単位の単語の総数である。従来N-gramとチャンクN-gramの場合、自立語のみだが、係受けN-gramは機能語も含む。"ideal" を計算した理由は、自立語のみの翻訳単位モデルに対して付属語を含めて被覆率を評価するのは公平でないと思われるからである。"cover" は被覆率を指す。接頭詞 "i_" は、"ideal" 中の正解対訳表現の単語数の割合を示し、"t_" は、コーパス中の全単語数中の正解対訳表現の割合を示す。1000文のテストデータに対して、英語側には14422個、日本語側には18998個の単語があった。"ideal" は閾値を下げる毎に増えていくのに対し、"total" は一定である。

f_{curr}	e	c	acc	e'	c'	acc'
100.0	0	0	n/a	0	0	n/a
50.0	0	0	n/a	0	0	n/a
25.0	1	1	1.000	1	1	**1.000**
12.0	2	2	1.000	3	3	**1.000**
10.0	5	5	1.000	8	8	**1.000**
9.0	4	4	1.000	12	12	**1.000**
8.0	3	3	1.000	15	15	**1.000**
7.0	6	6	1.000	21	21	**1.000**
6.0	9	9	1.000	30	30	**1.000**
5.0	17	16	0.941	47	46	**0.979**
4.0	31	31	1.000	78	77	**0.988**
3.0	64	64	1.000	142	141	**0.993**
2.0	349	256	0.733	491	397	**0.809**

表 2: 精度 (従来 N-gram)

f_{curr}	e	c	acc	e'	c'	acc'
100.0	2	2	1.000	2	2	**1.000**
50.0	2	2	1.000	4	4	**1.000**
25.0	10	10	1.000	14	14	**1.000**
12.0	32	32	1.000	46	46	**1.000**
10.0	9	9	1.000	55	55	**1.000**
9.0	14	14	1.000	69	69	**1.000**
8.0	21	21	1.000	90	90	**1.000**
7.0	17	16	0.941	107	106	**0.991**
6.0	18	16	0.888	125	122	**0.976**
5.0	38	35	0.921	163	157	**0.963**
4.0	93	91	0.978	256	248	**0.969**
3.0	138	134	0.971	394	382	**0.967**
2.0	547	518	0.946	941	900	**0.956**

表 3: 精度 (チャンク N-gram)

f_{curr}	e	c	acc	e'	c'	acc'
100.0	1	1	1.000	1	1	**1.000**
50.0	5	5	1.000	6	6	**1.000**
25.0	11	10	0.909	17	16	**0.941**
12.0	27	26	0.962	44	42	**0.955**
10.0	17	15	0.882	61	57	**0.934**
9.0	12	12	0.882	73	69	**0.945**
8.0	25	25	1.000	98	94	**0.959**
7.0	35	34	0.971	133	128	**0.962**
6.0	32	31	0.968	165	159	**0.964**
5.0	49	48	0.979	214	207	**0.967**
4.0	96	92	0.958	310	299	**0.965**
3.0	189	184	0.973	499	483	**0.968**
2.0	1003	818	0.815	1502	1301	**0.866**

表 4: 精度 (係受け N-gram)

5 考察

実験の結果、チャンク N-gram も係受け N-gram もそれぞれ従来 N-gram より良い結果が得られた。このことから、チャンク区切りと単語依存関係は、対訳表現抽出にいうて有用な言語情報であることがわかった。以下に、考察を行なう。

言語情報が一番少ない従来 N-gram とチャンク N-gram を比較すると、精度 (95%) の上でも被覆率 (40%) の上でも有効であることが判明した。従来 N-gram より、閾値 2 のとき、精度では 約 15%、被覆率では 10% の向上が観察された。チャンク区切り情報を考慮するだけで、大幅な精度と被覆率の向上が実現できた。従来 N-gram の方が、生成した翻訳単位数は多いが、実際に抽出された正解対訳表現は少なく、被覆率も低い。この現象の一説明としては、従来 N-gram は不必要な翻訳単位を生成してしまう傾向にあり、それが抽出アルゴリズムではノイズになっているのでは、と考えられる。このことから、チャンクで区切れば、翻訳単位生成の際の、わかち書きの曖昧性もなくなることがうかがえる。

次に、従来 N-gram と係受け N-gram を比較すると、閾値 2 のときに、抽出数、精度、被覆率のいずれにおいても係受け N-gram が優位である。従来 N-gram は自立語のみの翻訳単位であるが、係受け N-gram は機能語を含めた翻訳候補になっているため、翻訳単位の異なり数も対訳表現抽出数も多い。にもかかわらず、精度も上回っているのは、単語の依存関係は有効な言語情報であることが読みとれる。

表 8 に、各モデルのみで抽出できた正解対訳表現の例を示す。チャンク N-gram のみで抽出された対訳表現の大半は、固有名詞表現 (名詞句) で、1 対 1 対応がつきやすいものが多い。チャンク N-gram は、翻訳単位がチャンク区切りで閉じていることを考えると、この結果は直観的である。他の 2 つのモデルで対訳表現抽出されなかった理由としては、重なり合う翻訳単位を不必要に生成したためと思われる。抽出アルゴリズムで、重なりあう翻訳単位は対象外とするが、重なりあう翻訳単位が多く生成されるとノイズの原因になりやすい。

従来 N-gram モデルは、偶発的に共起した長めの対訳を抽出する傾向にある。これは、形態素解析が細かく区切っても、自立語のみを連結するためと考えられる。' look forward to visit/訪問を楽しみ' が多く出現しているため、従来 N-gram モデルでは、抽出された。チャンク N-gram では 'to' でチャンク区切りがあるために、この翻訳単位は生成されない。係受け N-gram では 'look forward/楽しみ'、' visit/訪問' は, 別々に抽出されたが、'look forward to visit' は抽出されなかった。

一方、係受け N-gram モデルでは機能語を含む対訳表現が抽出された。その中には、訳しわけのパター

f_{curr}	found(E)	ideal(E)	i_cover(E)	t_cover(E)	found(J)	ideal(J)	i_cover(J)	t_cover(J)
100.0	0	445	0	0	0	486	0	0
50.0	0	1182	0	0	0	1274	0	0
25.0	46	2562	0.018	0.0015	46	2564	0.018	0.0011
12.0	156	4275	0.036	0.0051	146	4407	0.033	0.0037
10.0	344	4743	0.073	0.0113	334	4935	0.068	0.0086
9.0	465	4952	0.094	0.0153	455	5247	0.087	0.0117
8.0	511	5242	0.097	0.0168	501	5593	0.090	0.0129
7.0	577	5590	0.103	0.0190	567	5991	0.095	0.0146
6.0	744	5944	0.125	0.0245	734	6398	0.115	0.0189
5.0	899	6350	0.142	0.0297	891	6894	0.129	0.0229
4.0	1193	6865	0.174	0.0394	1195	7477	0.160	0.0307
3.0	1547	7418	0.209	0.0511	1549	8257	0.188	0.0398
2.0	2594	8128	0.319	0.0857	2617	9249	0.283	0.0674

表 5: 被覆率 (従来 N-gram)

f_{curr}	found(E)	ideal(E)	i_cover(E)	t_cover(E)	found(J)	ideal(J)	i_cover(J)	t_cover(J)
100.0	92	253	0.364	0.0072	92	328	0.280	0.0092
50.0	122	764	0.160	0.0095	122	746	0.164	0.0122
25.0	243	1510	0.161	0.0191	236	1423	0.166	0.0236
12.0	439	2590	0.169	0.0345	432	2515	0.172	0.0432
10.0	483	2829	0.171	0.0379	472	2739	0.172	0.0472
9.0	540	3009	0.179	0.0424	526	2911	0.181	0.0526
8.0	629	3168	0.199	0.0494	623	3086	0.202	0.0623
7.0	687	3348	0.205	0.0540	681	3256	0.209	0.0681
6.0	760	3539	0.213	0.0597	754	3464	0.218	0.0754
5.0	871	3803	0.229	0.0685	864	3748	0.231	0.0864
4.0	1076	4091	0.263	0.0846	1070	4059	0.264	0.1070
3.0	1402	4409	0.318	0.1102	1391	4423	0.314	0.1391
2.0	2007	4803	0.418	0.1578	2004	4865	0.412	0.2004

表 6: 被覆率 (チャンク限定 N-gram)

f_{curr}	found(E)	ideal(E)	i_cover(E)	t_cover(E)	found(J)	ideal(J)	i_cover(J)	t_cover(J)
100.0	78	1454	0.054	0.0061	78	1957	0.040	0.0078
50.0	170	2495	0.068	0.0133	170	2715	0.063	0.0170
25.0	264	3787	0.070	0.0207	278	3606	0.077	0.0278
12.0	394	5470	0.072	0.0309	408	4465	0.091	0.0408
10.0	503	5947	0.085	0.0395	515	4709	0.109	0.0515
9.0	558	6192	0.090	0.0438	570	4837	0.118	0.0570
8.0	665	6456	0.103	0.0523	680	4967	0.137	0.0680
7.0	801	6788	0.118	0.0629	814	5123	0.159	0.0814
6.0	900	7110	0.127	0.0707	911	5274	0.173	0.0911
5.0	1043	7520	0.139	0.0820	1065	5449	0.195	0.1065
4.0	1249	8055	0.155	0.0982	1274	5674	0.225	0.1274
3.0	1690	8690	0.194	0.1329	1686	5992	0.281	0.1686
2.0	2665	9664	0.276	0.2095	2703	6531	0.414	0.2703

表 7: 被覆率 (係受け N-gram)

モデル	英語	日本語
従来	U.S.-Japan	日米
従来	look forward (to) visit	訪問 (-を-) 楽しみ
従来	give information	資料提供
チャンク	Hong Kong	香港
チャンク	San Diego	サンディエゴ
係受け	apply for position	職-に-応募-する
係受け	be at your service	用命-に-従い-ます
係受け	checking into matter	件-を-調査
係受け	tell about matter	件-について-お知らせ
係受け	free of charge	無料
係受け	out of question	問題-外
係受け	out of print	絶版

表 8: 各モデル特有の対訳表現の正解例

ンとして利用できるものがいくつか存在する。これは、機能語も含めて翻訳単位を生成したために抽出できたものと考える。

これらの考察から、自然言語処理ツールから得られるチャンク区切り情報は名詞句などを抽出するには、有用な言語情報であることが明らかになった。新分野の対訳を準備するときに有効である。しかし、イディオム表現などの長めの対訳表現抽出を目指すなら、係受け関係情報を利用すべきであることが窺える。

最後に、我々の手法の課題にふれる。それは、対訳コーパス中に一度しか同時出現しない対訳表現をいかに獲得するかという問題である。電子辞書を利用することも考えるが、北村らが行なった構造照合 [12] を検討する方向で考えてゆきたい。

6 まとめ

本稿では、対訳コーパスから対訳表現を自動抽出する際の翻訳単位の影響について比較実験を報告した。対訳表現抽出のために、N-gram モデルに基づく 3 つの翻訳単位モデルを提案し、日英の対訳コーパス 5000 文で実験を行なった。

本稿では、精度と被覆率の観点から 3 つのモデルの定性的な考察を行なった。結果、(1) 自然言語処理ツールから得られるチャンク区切り情報は対訳表現抽出において、有用な言語情報であり、新分野の対訳を準備するときに有効であること、(2) イディオム表現などの長めの対訳表現抽出では係受け関係情報が重要な役割をはたす、という知見が得た。対訳表現抽出において、チャンク区切りと単語の依存関係は、それぞれ有用で、相補しあう関係にある言語情報であると思われる。

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。本研究に関して有益な助言をいただいた沖電気工業の北村美穂子氏に感謝する。

参考文献

[1] *Text Chunking, CoNLL-2000 Shared Task Description.* http://lcg-www.uia.ac.be/conll2000/chunking/, 2000.

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

[3] M. Kitamura and Y. Matsumoto. Automatic extraction of word sequence correspondences in parallel corpora. In *Proc. of 4th WVLC*, pp. 79–87, 1996.

[4] 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.

[5] Y. Matsumoto and M. Asahara. Ipadic users manual. Technical report, 2001.

[6] I.D. Melamed. Models of translational equivalence. In *Computational Linguistics*, Vol. 26(2), pp. 221–249, 2000.

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

[8] 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.

[9] K. Yamamoto and Y. Matsumoto. Acquisition of phrase-level bilingual correspondence using dependency structure. In *COLING-2000*, pp. 933–939, 2000.

[10] 黒橋禎夫, 長尾眞. 京都大学テキストコーパス・プロジェクト. 言語処理学会第 3 回年次大会, pp. 115–118, 1997.

[11] 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.

[12] 北村美穂子, 松本裕治. 対訳コーパスを利用した翻訳規則の自動獲得. 情報処理学論文誌, 第 37(6) 巻, pp. 78–88, 1996.