大型英和辞典と関連用語収集技術を用いた対訳専門用語集の自動編集

佐々木　靖弘[†]　　佐藤　理史[‡]　　宇津呂　武仁[†]

[†]京都大学大学院情報学研究科　　[‡]名古屋大学大学院工学研究科
sasaki@pine.kuee.kyoto-u.ac.jp, ssato@nuee.nagoya-u.ac.jp, utsuro@i.kyoto-u.ac.jp

1. はじめに

ある特定の分野やトピックに関連する語を日英対訳の形で示したリスト（対訳用語集）は、いくつかの状況において、非常に有用な資料となる。たとえば、

- その分野やトピックに関連した英語のテキストを読む、書く、あるいは、翻訳する前の準備資料として。
- その分野やトピックに関連する英語のトークを聞く、あるいは、通訳する前の準備資料として。実際にトークを聞いているときに参照するメモ（備忘録）として。

一般的な状況では、知らない英単語が出てきた時、辞書を引いてその訳語を調べればよい。しかしながら、ある種の状況では、その場で辞書を引く時間的余裕がないため、前もって**予習する**ことが必要となる。また、時間的余裕がある「読む・書く・翻訳する」といった状況においても、比較的少数の重要な用語の訳語を前もって頭に入れておくことは、それらの作業をスムーズに進める効果がある。

対訳用語集を自動生成・編集する研究は、これまで、主に、訳語を推定することに焦点を当ててきた [1),2)]。機械翻訳への応用を想定した場合、辞書に載っていない訳語対を得ることが、ひとつの到達目標となる。しかしながら、実際に上記のような人間による利用を想定した場合、ほとんどの人は、英語と日本語の対応が 100%正しくない限り、その対訳用語集を使おうとしないであろう[*]。訳語の自動推定技術をそのようなレベルに持っていくことは、当面、絶望的である。

近年、大型の英和辞書がいくつか編纂され、電子的にも利用可能な状況となってきた。たとえば、グランドコンサイス [4)] のサイズは公称 36 万項目、英辞郎[**]のサイズは公称 143 万項目である。これらの収録語数は、典型的な人間が知っている語彙の数を遥かに越えており、特定の分野で使われる専門用語など、「辞書を引いて調べる語」が多数含まれていると考えるのが妥当である。つまり、これらの大型英和辞書に載っている語を拾うだけでも、冒頭に示したような、人間のための対訳用語集を作ることができる可能性がある。

[*] より正確に言うならば、(影浦 [3)] がよく指摘するように)、100%正しい**かのように**見なすことができる対訳用語集であればよい。

[**] http://www.eijiro.jp/

このような考え方に基づき、我々は、大型英和辞書から特定の分野の専門用語集を作成することを試みている。このアプローチは、辞書のエントリーを選択的に抜き出して、ある分野の用語集を作成することを実現する、いわば「情報の再生産」と位置付けることができる。本稿では、このようなアプローチの実現法と、大型英和辞書として英辞郎を利用した場合の実験結果について述べる。

2. 枠　　組

2.1 問題設定

ここでは、まず、対訳辞書から対訳用語集を作成する問題を、形式的に定義する。

対訳辞書 D を、対訳（対訳関係にある語の対）p の集合として定義する。

$$D : p \text{の集合} \tag{1}$$

作成すべき対訳用語集を T とする。どのような分野の対訳用語集を作成すべきかは、その分野を代表する、シード対訳 $p_s(\in D)$ として与えられるものとする。このとき、作成する対訳用語集を $T(p_s)$ と書くことにする。$T(p_s)$ は D の部分集合である。

$$T(p_s) \subset D \tag{2}$$

但し、$T(p_s)$ の要素の数は比較的少数とする。

これまで我々が行なってきた関連用語の自動収集の研究 [5)] に基づき、シード対訳 p_s と強く関連する対訳 p_i を集めたもので、$T(p_s)$ を構成する。まず、2 つの対訳間に関連度 $r(p_i, p_j)$ を定義する。この関連度が閾値 R より大きいものを集め、$T(p_s)$ とする。すなわち、

$$T(p_s) = \{p_i \mid p_i \in D \text{ and } r(p_s, p_i) \geq R\} \tag{3}$$

あるいは、閾値 R を用いずに、関連度上位 K 位までを集め、$T(p_s)$ を構成してもよい。

2.2 現実的な実現法

本研究で想定している D の要素の数は、数十万から百万超である。関連度 r がどの程度短時間で計算できるかどうかにもよるが、D のすべての要素に対して、p_s との関連度を計算するのは、現実的とは言えない。そのため、より計算時間がかからない方法で、$T(p_s)$ の要素となりそうな候補を集め、それらに対してのみ、実際に関連度を計算する方法をとる。$T(p_s)$ の要素となりそうな候補を集めた集合を $C(p_s)$ とし、これを作り出す操作を preselect と書くと、実際の計算手順は、次のようになる。

$$C(p_s) = \text{preselect}(D, p_s) \quad (4)$$

$$T(p_s) = \{p_i \mid p_i \in C(p_s) \text{ and } r(p_s, p_i) \geq R\} \quad (5)$$

2.3 ウェブテキストを利用した候補生成

先に作成した（日本語を対象とした）関連用語収集システムでも、関連度を実際に計算する候補語の集合を、関連度を計算せずに作り出すことが必要となり、ウェブテキストからシード語の周辺に現れる語を集めて候補語集合を作成する方法を採用した。本研究でもこれを踏襲する。なお、本論文ではこれ以降、対訳 p_i は、日本語の用語 j_i と英語の用語 e_i の対として扱う。

シード対訳 p_s の日本語側を j_s とする。ウェブテキストにおいて j_s の周辺に現れる用語集合を $N(j_s)$ とする。この集合には、j_s に強く関連する語が高密度で含まれることが期待できる。そこで、次のように候補集合 $C(p_s)$ を作成する考えが生まれる。

$$C(p_s) = \{p_i \mid p_i \in D \text{ and } j_i \in N(j_s)\} \quad (6)$$

ここで、$N(j_s)$ は、j_s の周辺のウェブテキスト $S(j_s)$ における頻度 $f(j_i, S(j_s))$ が閾値 F 以上の用語の集合とする。すなわち、

$$N(j_s) = \{j_i \mid f(j_i, S(j_s)) \geq F\} \quad (7)$$

一般に、集合 $N(j_s)$ には、基本語も含まれる。基本語は、本研究が目標とする対訳用語集には不要である。基本語を排除するような関連度を採用すれば、これは関連度によって排除されるが、計算効率の観点からは、あらかじめ排除しておくことが望ましい。そこで、基本語を集めた日本語辞書 B_J を仮定し、B_J に含まれるものは候補に含めないことにする。

$$C(p_s) = \{p_i \mid p_i \in D \text{ and } j_i \in N(j_s) \text{ and } j_i \notin B_J\} \quad (8)$$

ここまでの絞り込みは、すべて日本語側を対象とするものである。日本語の文章において、「英語由来の重要な用語には、英語表記が併記されることが多い」ということを利用すると、日本語コーパスを利用して英語用語を絞り込むことが可能となる。シード対訳の日本語用語 j_s から作成される分野コーパスを $J(j_s)$ とするとき、

$$C(p_s) = \{p_i \mid p_i \in D \text{ and } j_i \in N(j_s) \text{ and } j_i \notin B_J \text{ and } exist(e_i, J(j_s))\} \quad (9)$$

ここで、$exist(e_i, J(j_s))$ は、コーパス $J(j_s)$ に（何らかの意味で）e_i が出現することを表す。

2.4 対訳間の関連度

一方、対訳間の関連度 $r(p_i, p_j)$ は、多くの実現法が考えられる。我々は関連用語の自動収集と同様、ウェブヒット数を利用した Jaccard 係数を採用する。但し、その計算の対象は、日本語の用語ではなく、英語の用語とする。すなわち、

$$r(p_i, p_j) = Jac(e_i, e_j) \quad (10)$$

$$= \frac{hits(e_i \& e_j)}{hits(e_i) + hits(e_j) - hits(e_i \& e_j)} \quad (11)$$

英語用語間の関連度を用いる利点は、**対訳の曖昧性の解消**ができる点である。同一の日本語に対して複数の英訳が存在した場合に、関連度の高い方の英訳が、対象分野におけるその日本語の英訳としてより確からしいと判定することができる。

3. システム

以上の考えに基づいて、対訳辞書 D から、シード対訳 p_s が表す分野の対訳用語集 $T(p_s)$ を作成するシステムを実装した。システムは、以下の５つのステップにより構成される。

1. 候補語収集
2. 基本語フィルタ
3. 辞書引き
4. 英訳存在チェック
5. 関連度計算

この内、１～４のステップは、2.3 節で述べた候補集合 $C(p_s)$ を生成するための preselect の処理である。以下では、これらのステップの詳細について説明する。

3.1 候補語収集

候補集合 $C(p_s)$ は、ウェブテキストにおいて、シード対訳の日本語 j_s の周辺に現れる用語の集合 $N(j_s)$ から作成される。そこで本ステップでは、用語集合 $N(j_s)$ を以下の方法で収集する。

1. ウェブページ収集　「j_s とは」「j_s という」「j_s は」「j_s の」「j_s」をサーチエンジンのクエリとして入力し、検索結果の上位 100 ページずつを収集し、j_s を含むウェブページ $J(j_s)$ を得る。

2. テキスト抽出　$J(j_s)$ から、j_s を含む文、および、その前後２文を抽出し、テキスト $S(j_s)$ を作成する。

3. 候補語抽出　テキスト $S(j_s)$ を形態素解析し、$S(j_s)$ における頻度が F 以上の名詞および複合名詞を抽出し、$N(j_s)$ とする。

3.2 基本語フィルタ

$C(p_s)$ に含まれる対訳 p_i の日本語 j_i は、基本語日本語辞書 B_J には含まれない。そこで本ステップでは、前節で収集した $N(j_s)$ から基本語辞書 B_J に含まれる j_i を除外する。

3.3 辞書引き

候補集合 $C(p_s)$ に含まれる対訳 p_i の最も基本的な条件は $p_i \in D$ （p_i が対訳辞書 D に含まれること）である。そこで、本ステップでは候補語フィルタにより数を絞り込まれた $N(j_s)$ 中の日本語用語 j_i の英訳 e_i を対訳辞書 D から求める。この処理は、ここまでのステップの任意の時点で行なうことが可能であるが、ここまでの処理は日本語に対してのみ行なってきたので、システム実装上の観点から、この時点で行なうことにする。

この処理により、(8) 式に示す $C(p_s)$ が得られたことになる。

3.4 英訳存在チェック

preselect の最後の処理は、日本語の分野コーパス $J(j_s)$ に、e_i が存在する対訳 p_i に候補集合 $C(p_s)$ を絞り込む処

理である。すなわち、ここでは、(9) 式の $exist(e_i, J(j_s))$ を具体的に定義する。$exist(e_i, J(j_s))$ の定義として、最も厳格なのは、e_i がそっくりそのまま $J(j_s)$ に現れるとき、$exist(e_i, J(j_s))$ が成り立つとする定義である。すなわち、

$exist(e_i, J(j_s))$ ：e_iが $J(j_s)$ にフレーズとして存在する (12)

しかし、実際にいくつかの入力に対して実験を行なったところ、(12) 式の定義では、候補対訳を絞り込み過ぎてしまう結果となった。そこで、(12) 式を少し緩めて、次のような定義を $exist(e_i, J(j_s))$ の定義として採用した。

$exist(e_i, J(j_s))$ ：e_iの構成単語がすべて $J(j_s)$ に存在する (13)

すなわち、本ステップでは、構成単語のいずれか一つでも $J(j_s)$ に含まれない e_i を持つ対訳 p_i を候補集合 $C(p_s)$ から除外する。日本語の分野コーパス $J(j_s)$ としては、3.1 節の候補語収集で収集したウェブページを用いる。この処理により、(9) 式に示す候補集合 $C(p_s)$ が得られる。

3.5 関連度計算

最後に、候補集合 $C(p_s)$ 中のそれぞれの対訳 p_i とシード対訳 p_s との関連度 $r(p_s, p_i)$ を計算し、(5) 式を満たす対訳用語集 $T(p_s)$ を得る。$r(p_s, p_i)$ としては、2.4 節で述べたように、各対訳の英語 e_s と e_i のウェブヒット数を利用した Jaccard 係数を用いる。

出力の $T(p_s)$ では、同一の日本語 j_i に対して、複数の英訳が存在する場合がある。このような場合は、関連度が高い方の英訳を持つ対訳を優先的に提示する。

4. 実　　験

作成したシステムに、表 1 の 6 つの対訳をシード対訳として入力し、対訳用語集作成実験を行なった。

本実験の評価は、翻訳者・通訳者が各分野の英語テキストを読んだり、英語のトークを聞いたりする際に準備する対訳リストとして有用であるかどうか、という観点で行なうことが望ましい。しかし、そのためには、翻訳のプロ、あるいは、各分野の専門家の知識が必要となり、非常にコストがかかる。そこで、本論文では、当座の評価法として、各シード対訳の日本語 j_s に対して参照セットと呼ばれる用語セットを作成し、この参照セットに基づいて評価を行なうことにした。

参照セットは以下の手順で作成した。

1. 用語 j_s が表す専門分野について書かれた書籍を 3 冊用意する。
2. それぞれの書籍から巻末の索引語をすべて収集し、参照セット $Ref(j_s)$ とする。

実験は、以下の手順に従う。

1. 表 1 の対訳の一つをシード対訳 p_s としてシステムに入力し、対訳用語集 $T(p_s)$ を得る。
2. $T(p_s)$ 中の各対訳 p_i の日本語 j_i に、参照セット $Ref(j_s)$ の用語がどれだけ含まれるかをカウントする。

対訳辞書 D としては英辞郎（ver.91）を、日本語基本語辞書 B_J としては岩波国語辞典（第 5 版）[6] を用いた。また、日本語サーチエンジンとしては goo*を、英語サーチエンジンとしては Google Web APIs**を用いた。(6) 式の頻度閾値は $F = 2$、(5) 式の関連度閾値は $R = 0.01$ に設定した。

結果を表 1 の N、C、T、および Ref に示す。N は、ウェブテキストにおいて j_s の周辺に現れる日本語 $N(j_s)$ の用語数、C は、候補対訳集合 $C(p_s)$ に含まれる対訳数、T は、システムが出力した対訳用語集 $T(p_s)$ に含まれる対訳数、Ref は、$T(p_s)$ の各対訳の日本語の内、参照セット $Ref(j_s)$ に含まれる用語数である。

また、シード対訳として表 1 の p_{s1}（自然言語処理、natural language processing）を入力したときにシステムが出力する対訳用語集 $T(p_{s1})$ を、表 2 に示す。この表において、$r(p_s, p_i)$ は (11) 式で示した、シード対訳と候補対訳の関連度である。「※曖昧」欄は、各対訳の日本語 j_i に対して複数の英訳が存在したときに、対訳として選ばれた e_i の次に関連度が高かった英訳である。丸括弧内はその関連度を示す。例えば、「音声認識」の英訳としては “voice recognition” より “speech recognition” の方が「自然言語処理」分野の英訳としては確からしい（関連度が高い）ことを示している。なお、「参照」欄に $\surd$ がある対訳は、j_i が参照セット $Ref(j_s)$ に含まれる対訳である。

考察

表 1 に示すように、シード対訳 p_s に対してウェブコーパスから収集される $N(j_s)$ のサイズは、500〜600 程度である。この集合から、基本語フィルタ、辞書引き、英訳存在チェックの 3 つのフィルタリング処理により、50〜60(10%) 程度に絞り込まれた候補語集合 $C(p_s)$ が得られている。そして、最終的に得られる対訳集合 $T(p_s)$ のサイズは、20 程度である。$T(p_s)$ 中の各対訳の日本語の内、参照セットに含まれる用語は、その半数に満たない場合がほとんどである。

しかし、実際に収集された対訳集合を眺めると、直感的にはさほど的外れの用語が集まっているとは思われない。例えば表 2 の場合、「自然言語処理」分野から少し離れた分野の用語（「パターン認識」など）も含まれるが、参照セットには含まれない用語であっても、「対話システム」などの「自然言語処理」に強く関連する用語が含まれている。p_{s1} 以外の分野に対しては、我々は専門家ではないので、確定的なことは言えないが、主観的な印象としては、$T(p_{s1})$ と同様に、参照セットの用語以外にも、関連が強い用語が含まれているように思われる。

---

* http://www.goo.ne.jp/
** http://www.google.com/apis/

表 1 シード対訳と対訳専門用語集作成結果

p_s	j_s	e_s	N	C	T	Ref
p_{s1}	自然言語処理	natural language processing	573	50	22	9
p_{s2}	情報理論	information theory	529	59	23	5
p_{s3}	パターン認識	pattern recognition	597	54	13	11
p_{s4}	バイオインフォマティクス	bioinformatics	973	80	17	4
p_{s5}	マクロ経済学	macroeconomics	548	64	26	11
p_{s6}	ミクロ経済学	microeconomics	509	48	17	6

表 2 $T(p_{s1})$

$r(p_s, p_i)$	j_i	e_i	※曖昧	参照
0.059	言語処理	language processing		
0.072	情報抽出	information extraction		✓
0.062	機械翻訳	machine translation		✓
0.062	自動翻訳	machine translation		
0.043	自然言語理解	natural language understanding		
0.032	意味解析	semantic analysis		✓
0.030	音声認識	speech recognition	voice recognition (0.004)	✓
0.028	音声処理	speech processing	voice proessing (0.003)	
0.021	形態素解析	morphological analysis		✓
0.019	パターン認識	pattern recognition		
0.016	文脈自由文法	context-free grammar		✓
0.014	構文解析	parsing		✓
0.013	機械翻訳システム	machine translation system		
0.012	対話システム	dialogue system		
0.012	統計的手法	statistical approach		
0.012	言語データ	language data		✓
0.012	コーパス	corpus		✓
0.011	自然言語処理システム	natural language processing system		
0.011	データマイニング	data mining		
0.010	検索技術	search technology		
0.010	文字認識	character recognition		
0.010	意味論	semantics		

5. おわりに

本論文では、大型の対訳辞書から、特定の分野の専門用語集を作成する方法を示した。大型辞書を用いることによって、(対訳の正しさという面において) 実際に人間が使うことができる対訳集を作成することが可能である。

大型対訳辞書として英辞郎を用いて、日英専門用語対訳集を作成するシステムを実装し、実験を行なった。専門分野について書かれた書籍の索引語を用いた評価では十分に対象分野の対訳集合を収集できたとは言えないが、主観的な印象では、このようなアプローチの最初の試みとして、十分に期待を抱くことのできる対訳集合を集めることができている。

今回は対訳間の関連度として英語用語間の関連度を用いたが、今後は、日本語用語間の関連度も考慮に入れて、対訳間の関連を捉えることを考えている。また、より高速に処理を行なえるようにすることも、今後の課題である。

本研究の一部は、次の研究費による；科学研究費補助金・特定領域研究「実世界の関連性を投影した語彙空間の構築」(課題番号 16016249)、科学研究費補助金・基盤 (A) 「翻訳者を支援するオンライン多言語レファレンス・ツールの構築」(課題番号 17200018)。

参 考 文 献

1) 外池昌嗣, 木田充洋, 高木俊宏, 宇津呂武仁, 佐藤理史: 要素合成法を用いた専門用語の訳語候補生成・検証, 言語処理学会第 11 回年次大会発表論文集, pp. 17–20 (2005).
2) 木田充洋, 宇津呂武仁, 日野浩平, 佐藤理史: ウェブ上の日英非対訳文書を用いた訳語対応推定, 言語処理学会第 10 回年次大会発表論文集, pp. 253–256 (2004).
3) 影浦峡, 佐藤理史, 竹内孔一, 宇津呂武仁, 辻慶太, 小山照夫: 翻訳者支援のための言語レファレンス・ツール高度化方針, 言語処理学会第 12 回年次大会発表論文集 (2006).
4) 三省堂編修所 (編): グランドコンサイス英和辞典, 三省堂 (2001).
5) 佐々木靖弘, 佐藤理史, 宇津呂武仁: 用語間の関連度を測る指標の提案, 言語処理学会第 10 回年次大会発表論文集, pp. 25–28 (2004).
6) 西尾実, 岩淵悦太郎, 水谷静夫 (編): 岩波国語辞典第 5 版, 岩波書店 (1994).