英語ウィキペディアを日本語で引く：性能向上の検討

岡田 昌也　　佐藤 理史　　駒谷 和範
名古屋大学大学院 工学研究科 電子情報システム専攻
{masaya_o, ssato, komatani}@nuee.nagoya-u.ac.jp

1　はじめに

現在、語（ターム）の意味を調べるために、Web 上の百科事典『ウィキペディア』が広く用いられている。ウィキペディアには、多くの言語の版があるが、その中で、規模が最大の英語版（EnWiki）は、日本語版（JaWiki）の約 5 倍の記事数を有する。

日本語話者が、日本語タームの意味を調べる場合、JaWiki に求める記事があれば、それが一番便利である。しかし、そのような記事がない場合、EnWiki の記事は次善の策となる。ある程度の英語力があれば、その英語記事から必要な情報を得ることができる。

ここで、次のような問題が浮かび上がる。それは、「どうやって EnWiki を引くか」という問題である。EnWiki を引くためには、調べたい日本語タームの英訳が必要であるが、その英訳は、その時点では判明していないのが普通である。

我々は、上記の問題を解決するために、EnWiki を日本語で引くことができるシステムを提案している [岡田 11][Sato 11]。本システムの中核は、非生産型機械翻訳（Non-Productive Machine Translation, NPMT）を用いて入力タームを英訳する機能である。NPMT は、翻訳すべきタームの訳語を、大規模な訳語候補集合の中から選択する翻訳方式で、本システムでは、候補集合として EnWiki の見出し語リストを用いる。

これまで、我々は、NPMT を日英翻訳に適用するために、いくつかの拡張機構を導入してきた [Sato 11]。さらなる性能向上を目指し、今回、新たな拡張機構を導入する。さらに、新しく作成したテストセットを使用して、システムの性能を評価する。

以下、2 節で NPMT について述べ、3 節では、新たに導入した拡張機構について説明する。4 節では、テストセットを使用した本システムの評価について述べる。

2　非生産型機械翻訳

非生産型機械翻訳は、タームの翻訳に特化した翻訳方式で、あらかじめ大規模な訳語候補集合 (ターゲットリスト) T を準備しておき、この中から、翻訳すべきターム (ソースターム) s の訳語を選択する [岡田 10]。この方式では、翻訳のための知識源として、対訳辞書 D (訳語対の集合) を利用する。例えば、日英対訳辞書では、$\langle$ 電気双極子, electric dipole$\rangle$ のような訳語対 (日本語の文字列と英語の単語列のペア) がその要素となる。

本方式では、訳語対の列 $\delta \in D^*$ が、より大きな訳語対、および、その部分対応を表現する。例えば、

$$d_1 = \langle \text{電気双極子}, \text{electric dipole} \rangle$$
$$d_2 = \langle \text{放射}, \text{radiation} \rangle$$

とするとき、訳語対の列 $\delta = d_1 d_2$ は、

$$\begin{aligned}\delta &= d_1 d_2 \\ &= \langle \text{電気双極子}, \text{electric dipole} \rangle \langle \text{放射}, \text{radiation} \rangle \\ &= \langle \text{電気双極子放射}, \text{electric dipole radiation} \rangle\end{aligned}$$

という 3 つの情報を同時に表わす。ここで、δ のソース側とターゲット側を、それぞれ、$\mathrm{src}(\delta)$ と $\mathrm{tgt}(\delta)$ で表すことにする。

ソースターム s が与えられると、本方式は、

$$\text{(ソース側条件)} \qquad \mathrm{src}(\delta) = s \tag{1}$$
$$\text{(ターゲット側条件)} \qquad \mathrm{tgt}(\delta) \in T \tag{2}$$

の 2 つの条件を満たす訳語対 $\delta \in D^*$ をすべて求め、そのターゲット側 $\mathrm{tgt}(\delta)$ を、s の訳語として出力する。出力される訳語は 1 個とは限らず、複数の場合も、0 個の場合もある。

3　日英ターム翻訳への適用

いま、日本語ターム j から対応する英語ターム e を求めることを考えよう。理論的には、(1) 対訳辞書 D

が訳語対 $\langle j, e \rangle$ を生成可能であり、かつ、(2) e がターゲットリスト T に含まれていれば、前節の方式で j から e を必ず得ることができる。しかしながら、現実には、しばしば (1) が満たされないことがある。これは、現実に利用できる D が、理論上必要となる部分訳語対を、網羅的には収録していないためである。

この問題に対し、これまで、我々は辞書引きを拡張する方法で対処してきた [Sato 11]。しかしながら、システムの性能を評価した結果、依然として、ある種の訳語対の不足が補えていないことがわかった。このため、今回、新たに 2 つの拡張機構を導入した。本節では、まず、3.1 節で、どのような訳語対の不足が補えていなかったのかを説明し、3.2 節で、新たに導入した拡張機構について述べる。

3.1 ターム翻訳に見られる現象

3.1.1 表記ゆれ

日本語では、語の表記が複数存在する場合がある。タームの翻訳において特に顕著なのは、次の 3 種類の表記ゆれである。

1. 数字の表記ゆれ (漢数字と算用数字)
   例:)「二 名式命名法」と「2 名式命名法」
2. カタカナの表記ゆれ
   例:)「モホロ ビ チッチ」と「モホロ ヴィ チッチ」
3. 漢字の表記ゆれ
   例:)「浸 蝕」と「浸 食」、「蛋白」と「タンパク」

対訳辞書 D の検索は、表記 (文字列) で行なうので、一方の表記のエントリーしか辞書に存在しない場合、もう一方の表記で検索した場合は、検索に失敗する (訳語を得ることができない)。このため、すべての表記を収録した辞書が必要であるが、現実の辞書は、表記ゆれをカバーしていないことが多い。

これまで、この問題に対しては、辞書引きの段階で、辞書を引く語に表記変換ルールを適用する方法で対処してきた。変換ルールのうち、漢字の表記ゆれを対象としたルールは、『表記統合辞書』[1] から作成していたが、そのカバー率は、十分ではなかった。

3.1.2 翻訳ゆれ

ある英語は、日本語訳が複数存在する場合がある。この現象を、本研究では、翻訳ゆれと呼ぶことにする。特に顕著なのは、次の 2 種類の翻訳ゆれである。

1. 日本語の単語とトンランスリタレーション
   例:) orbital → 軌道 / オービタル
2. 異なる日本語の単語（同義語）
   例:) proprioceptor → 自己受容器 / 自己受容体

対訳辞書は、このような翻訳ゆれもカバーしていないことが多い。本研究では、これまで、同義語の翻訳ゆれに対しては、全く対処を行っていなかった。

3.2 日英ターム翻訳のための拡張

前節で述べた 2 つの問題に対処するために、辞書引きの段階において、新たに 2 つの拡張機構を導入する。本節では、翻訳すべきターム j の部分文字列を s で表す。辞書引きは、j のあらゆる部分文字列 s に対して実行される。

3.2.1 読みで辞書を引く

ルールでカバーできない表記ゆれのうち、漢字の表記ゆれに対しては、読みで辞書を引く方法で対処する[2]。まず、事前準備として、形態素解析器を用いて、辞書中のすべての見出し語に、読みを付与しておく。辞書引きの際には、s も読みに変換し、読みの一致するエントリーを検索する。この方法は、読みが同じものをすべて検索するため、誤った訳語を得てしまう可能性も高い。

3.2.2 類似文字列検索ツールの導入

同義語の翻訳ゆれに対しては、類似文字列検索ツールを導入し、s の類似文字列 s' で辞書を引く方法で対処する。このような方法を用いるのは、「自己受容器」と「自己受容体」のように、同義語は、共通文字をもつことが多いためである。

類似文字列検索は、s に対して、誤った訳語を得てしまう可能性が非常に高い。このため、合成操作によって得られる j に対する訳語（システムの出力）e も、誤りである可能性が高い。そこで、この方法を用いて出力が得られた場合には、サーチエンジンを用いて、適切なもの選択するという後処理を追加する。具体的には、e と j のアンドヒット数を求め、その数が 1 以上の e のうち、上位 3 位までを選択する。

3.3 拡張の優先順位

最終的に用いる拡張機構は、従来からの 5 つの拡張機構に、上記の 2 つの拡張機構を加えた 7 つとなる。一般に、拡張機構の導入により、出力される訳語の数は増加する。その一方で、誤った訳語を出力する危険性も高くなる。そのため、比較的安全な拡張機構から優先して使用し、訳語が一つも得られない場合のみ、より危険な拡張機構を使用する。表 1 に拡張レベルと、それぞれのレベルで使用する拡張機構を示す。この表の拡張機構 1〜5 は、従来からの拡張機構である。

[1] http://www2.ninjal.ac.jp/lrc/index.php

[2] この方法は、文献 [岡田 11] で一旦採用したが、文献 [Sato 11] では削除されている。

表 1: 拡張レベル

拡張機構	L_1	L_2	L_3	L_4
1. 無翻訳	√	√	√	√
2. トランスリタレータ	√	√	√	√
3. 表記変換ルール	√	√	√	√
4. ダミー入力	√	√	√	√
5. 接尾辞・付属語		√	√	√
6. 読み			√	√
7. 類似文字列				√

4 実験

本節では、作成したシステムを評価するための実験と結果について述べる。

4.1 使用したデータおよびツール

実験では、次のデータおよびツールを使用した。

1. 対訳辞書 D
   英和辞書『英辞郎』ver.116 から作成した。具体的には、『英辞郎』に収録されている訳語対と、それらから抽出した部分訳語対 [外池 07] を併せたものを用いた。そのサイズは、2,366,649 ペア (日本語: 1,507,143; 英語: 1,887,518) である。
2. ターゲットリスト T
   EnWiki の見出し語リスト (5,907,150 件) を用いた。
3. トランスリタレータ
   『絣』[Sato 10] を用いた。
4. 類似文字列検索ツール
   『SimString』[岡崎 11] を用いた。
5. 読みを求めるための形態素解析器
   MeCab[3] + UniDic [4] を用いた。

4.2 テストセット

テストセットは、以前からの Ox に加え、Jes と Mix の 2 つを新たに作成した。

1. オックスフォードテストセット Ox
   『オックスフォード科学辞典』[Daintith 09] から、次の 3 つの条件を満たす訳語対 $\langle j, e\rangle$ を収集し、Ox として用いた。

   $\overline{W_j}$: j は JaWiki の見出し語でない
   W_e: e は EnWiki の見出し語である
   H: j は Web 上に 10 回以上出現する

   Ox のサイズは、2,534 である。
2. 日・英・西テストセット Jes
   『日・英・西 技術用語辞典』[小谷 90] から、上記の 3 条件を満たす日英の訳語対を収集し、Jes として用いた。Jes のサイズは、6,036 である。
3. 分野混合テストセット Mix
   このテストセットは複数の辞書から作成した。まず、23 種類の『学術用語集』[5] と『経済・法律 英和・和英辞典』[尾崎 06] から、上記の 3 条件を満たす訳語対を収集し、25 分野の訳語対集合を作成した。ここで、25 分野なのは、「経済・法律英和・和英辞典」から収集された訳語対を、経済分野と法律分野に分類したためである。次に、各分野から、100 ペアを無作為に抽出し、それらを合わせて Mix として用いた。Mix のサイズは、2,500 である。

このうち、テストセット Ox は、システムの開発（3 節の拡張機構の導入）に使用した。

4.3 実験結果と検討

テストセットに含まれるそれぞれのペア $\langle j, e\rangle$ に対し、その日本語側 j を入力したとき、どのような出力が得られるかを調べた。出力されたそれぞれの訳語 e' は、EnWiki において、e と e' が同一記事を指し示す場合に正解と判定した。それぞれの入力 j に対する判定は、次の 4 種類に分類した。

Perfect: 正解訳語のみを出力
Ambiguous: 正解訳語の他に、不正解訳語を出力
False: 不正解訳語のみを出力
None: 出力なし

実験結果を、表 2 に示す。

この表の最初の 7 行は、ベースラインと提案方式の性能を示している。このうち、1 行目と 2 行目は、ベースラインに相当し、翻訳に『英辞郎』単体を用いた場合（1 行目）、『英辞郎』とターゲットリスト T による訳語限定を組み合わせた場合（2 行目）の性能を表す。3 行目以降の NPMT_n は、提案方式の性能を表す。ここで、添字 n は、使用する拡張レベルを最大 L_n に制限することを意味する。なお、NPMT_0 は、拡張を一切用いないことを表す。表の数値は、P、あるいは、A、F、N と判定された割合（入力数 / テストセットサイズ）をパーセントで示している。なお、P+A は、正解訳語を出力できた割合を示す。

英辞郎と英辞郎 w/T の差分は、T の効果を表す。いずれのテストセットに対しても、Ambiguous が減少し、その代わりに Perfect が増加していることから、T の使用には、訳語候補の数を抑制し、Perfect の割合を高める効果があることがわかる。

[3] http://mecab.sourceforge.net/
[4] http://www.tokuteicorpus.jp/dist/
[5] 文部省が編纂した用語集で、「建築学編」などがある

表 2: 実験結果

テストセット	*Ox* (2,534 対)					*Jes* (6,036 対)					*Mix* (2,500 対)				
	P	A	P+A	F	N	P	A	P+A	F	N	P	A	P+A	F	N
英辞郎	25.1	21.1	46.2	13.8	40.0	15.4	32.7	48.1	23.0	28.9	22.4	27.1	49.5	17.3	33.3
英辞郎 w/*T*	38.8	7.4	46.2	5.9	47.9	30.5	17.6	48.1	12.5	39.4	38.9	10.5	49.5	8.6	42.0
$NPMT_0$	41.8	14.8	56.6	8.7	34.7	30.7	30.6	61.3	15.4	23.3	37.8	23.8	61.6	11.7	26.6
$NPMT_1$	45.7	19.1	64.8	9.2	26.0	31.0	33.7	64.7	16.4	18.9	38.6	26.9	65.5	12.3	22.2
$NPMT_2$	49.2	20.6	69.8	11.1	19.1	31.4	34.1	65.5	17.4	17.2	40.0	27.6	67.6	13.6	18.8
$NPMT_3$	49.9	21.0	70.9	12.4	16.8	32.3	35.0	67.3	19.2	13.6	40.6	28.4	69.0	16.3	14.7
$NPMT_4$	50.6	26.4	77.0	20.4	2.7	32.6	38.1	70.7	27.9	1.4	41.1	31.3	72.4	25.0	2.6
Google w/*T*	61.3		–	11.6	27.0	48.3		–	26.8	24.9	60.4		–	16.8	22.8
Transer w/*T*	62.9		–	9.4	27.7	50.9		–	21.8	27.3	63.3		–	14.3	22.4
$NPMT_4$+	69.5		–	27.9	2.7	56.3		–	42.7	1.0	63.6		–	34.4	2.0

英辞郎 w/*T* と $NPMT_0$ の差分は、訳語対を合成する効果を表す。いずれのテストセットに対しても、P+A が 10%程度向上していることから、ターム翻訳が、単に対訳辞書を引く方法では不十分であり、訳語対の合成操作が必要であることがわかる。

$NPMT_0$〜$NPMT_4$ の比較では、それぞれの拡張レベルで、どの程度性能が向上するのかがわかる。新たに導入した拡張機構を使用する $NPMT_3$ と $NPMT_4$ においては、いずれのテストセットに対しても、P+A の増加が見られた。このことから、導入した拡張機構は、有効に機能していると言える。

拡張の有効性には、テストセットによって差が見られた。$NPMT_0$ と $NPMT_4$ を比較すると、*Ox* に対しては P+A が 20%程度向上しているのに対し、*Jes* と *Mix* に対しては 10%程度しか向上していないことがわかる。*Ox* に対して拡張が最も有効なのは、前述したように、*Ox* を使用してシステム開発を行ったためである。本方式のさらなる性能向上を果たすには、今後、*Jes* と *Mix* に対する結果を分析する必要がある。

表 2 の最後の 3 行は、本方式の性能を他の翻訳システムと比較した結果である。ここでは、『Google 翻訳』[6]および、市販の翻訳ソフト『MAC-Transer 2010』[7]と比較した。これら 2 つの翻訳システムは、1 件の入力に対し、高々1 件の訳語しか出力しない。これらのシステムと公平な評価を行なうために、本方式が訳語を 2 件以上出力した場合は、サーチエンジンを利用して入力と各訳語とのアンドヒット数を求め、この数が最も多かったものを最終的な出力とする後処理を追加した ($NPMT_4$+とする)。さらに、$NPMT_4$+以外の翻訳システムに対し、ターゲットリスト *T* を用いて、訳語を限定する処理を追加した。

Perfect の割合に注目すると、いずれのテストセットに対しても、$NPMT_4$+の結果が最も良かった。このことから、正解訳語を出力できるかという点において、本方式は、最も優れていると言える。なお、テストセット *Ox* と *Jes* に対する性能の差は、統計的に有意 (α=0.05) であるが、*Mix* に対する性能の差は、統計的に有意ではない。

一方、False の割合に注目すると、3 つのシステムの中で、$NPMT_4$+が最も大きい。これは、要素合成、および、拡張機構が、正解訳語の生成に貢献する一方で、不正解訳語の生成も促進してしまうためである。今後、この不正解訳語の生成を抑制する方法を検討していく必要がある。

謝辞 本研究は、Google Inc. の University Research Program for Google Translate 、および、科学研究費補助金挑戦的萌芽研究 (課題番号 22650047) の支援を受けている。

参考文献

[Daintith 09] Daintith, J. ed., 山崎 昶 (訳) : オックスフォード科学辞典, 朝倉書店 (2009)

[小谷 90] 小谷 卓也, 郡 亜都彦 : 日・英・西 技術用語辞典, 研究社 (1990)

[尾崎 06] 尾崎 哲夫 : 経済・法律 英和・和英辞典, ダイヤモンド社 (2006)

[外池 07] 外池 昌嗣, 宇津呂 武仁, 佐藤 理史 : ウェブから収集した専門分野コーパスと要素合成法を用いた専門用語訳語推定, 自然言語処理, Vol. 14, No. 2, pp. 33–68 (2007)

[Sato 10] Sato, S.: Non-productive machine transliteration, in *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pp. 16–19 (2010)

[岡田 10] 岡田 昌也, 佐藤 理史 : 大規模訳語候補集合を利用した専門用語翻訳, 第 24 回人工知能学会全国大会論文集, 2C4-1 (2010)

[岡田 11] 岡田 昌也, 佐藤 理史 : 英語ウィキペディアを日本語で引く, 第 25 回人工知能学会全国大会論文集, 2F1-1 (2011)

[Sato 11] Sato, S. and Okada, M.: Japanese-English Cross-Language Headword Search of Wikipedia, Proceedings of the 9th International Conference on Terminology and Artificial Intelligence, pp. 45–51 (2011)

[岡崎 11] 岡崎 直観, 辻井 潤一 : 集合間類似度に対する簡潔かつ高速な類似文字列検索アルゴリズム, 自然言語処理, Vol. 18, No. 2, pp. 89–118 (2011)

[6] `http://translate.google.co.jp/#`, 2012/1/18 現在
[7] `http://www.crosslanguage.co.jp/products/mac2010/`