Bilingual KWIC—対訳表現抽出の可視化による翻訳支援

小川 泰弘^{†,††}・外山 勝彦^{†,††}

計算機による対訳表現抽出を可視化することにより、対訳辞書の構築や翻訳を支援するツール Bilingual KWIC を開発した。本ツールは、入力されたキーワードに対する対訳表現を自動的に推定し、それらを含む原言語文と対象言語文をそれぞれ KWIC 形式で表示することにより、ユーザの翻訳作業などを支援する。技術的には、形態素解析などを利用せずに文字列情報だけから対訳を抽出するため、どのような言語対にも適用可能であり、さらには単語以外の表現に対しても対訳を表示することが可能である。また対訳表現を KWIC 形式で表示することにより、システムの抽出誤りに対する修正を容易にするだけでなく、派生表現の獲得や複数の対訳表現の比較も可能としている。本稿では、Bilingual KWIC の特徴と開発経緯について述べる。
キーワード：翻訳支援、対訳辞書構築、バイリンガル・コンコーダンサ、Dice 係数

Bilingual KWIC—GUI Translation Support Tool Based on Bilingual Expression Extraction

YASUHIRO OGAWA^{†,††} and KATSUHIKO TOYAMA^{†,††}

This paper proposes a GUI support tool for bilingual dictionary compilation and translation, called “Bilingual KWIC.” Bilingual KWIC acquires bilingual expressions from a parallel corpus and displays the result in KWIC format. Displaying in KWIC format enables users to easily correct errors of word alignment and compare two or more types of equivalents. Since Bilingual KWIC does not use morphological information and uses only character-level information, it can deal with any language pairs and can acquire translation equivalents for any input other than words. In this paper, we introduce Bilingual KWIC and describe its features and development process.

Key Words: *Translation Support Tool, Bilingual Dictionary Compilation, Bilingual Concordancer, Dice Coefficient*

1 はじめに

機械翻訳システムの性能向上や大量のコーパスを伴う翻訳メモリなどの導入により、機械支援翻訳 (CAT) が広く行われるようになってきている。その一方で、翻訳の対象となる文書の内容が専門的である場合、その分野特有の専門用語や定型表現に関する対訳辞書が必要となる。

[†] 名古屋大学情報基盤センター, Information Technology Center, Nagoya University

^{††} 名古屋大学大学院情報科学研究科, Graduate School of Information Science, Nagoya University

そうした辞書を人手で作成することはコストが高いため、あらかじめ翻訳された対訳コーパスから専門用語や定型表現の対訳を自動抽出する研究が盛んである (Matsumoto and Utsuro 2000). しかし、自動抽出の結果は必ずしも正確ではなく、間違った対訳表現を抽出したり、対訳表現の一部だけを抽出する場合がある。また、一つの語に対して複数の対訳表現を抽出した場合に、訳し分けに関する知見が必要となる。

そこで、対訳表現を抽出するだけでなく、対訳表現の各候補を、それが出現した文脈と一緒に表示することによって、ユーザによる対訳表現の選定を支援し、対訳辞書構築を支援するシステム Bilingual KWIC[®] を開発した。Bilingual KWIC は、対訳抽出の技術と KWIC (KeyWord In Context) 表示 (Luhn 1960) を統合し、文単位で対応付けされたパラレル・コーパスから、与えられたキーワードとその対訳表現の候補をそれぞれ文脈付きで表示する。

Bilingual KWIC の開発過程については 5 章において詳しく述べるが、最初は法律分野の対訳辞書構築を支援する目的で開発した。しかし、このシステムは対訳辞書構築だけでなく、翻訳支援にも有用であるため、その後に関発された法務省・日本法令外国語訳データベース・システム (JLT)¹ (外山, 齋藤, 関根, 小川, 角田, 木村, 松浦 2012) においても採用されるに至った。JLT は、日本の主要法令とその英訳、法令用語日英標準対訳辞書、および日本法令の英訳に関する関連情報をインターネット上において無償で提供するウェブサイトである。また、Bilingual KWIC は名古屋大学が開発した学内情報翻訳データベース NUTRIAD² (福田, 外山, 野田 2013) でも採用され、学内文書の英文化を支援し、大学の国際化に寄与している。NUTRIAD のシステムは、九州大学・熊本大学・東北大学でも導入され、Bilingual KWIC も同様に利用されている。

Bilingual KWIC の現在の目的は、対訳辞書のようにあらかじめ登録された訳語と少数の用例を提示するのではなく、任意の入力キーワードに対して対訳表現を計算し、豊富なパラレル・コーパスからの情報を一緒に提示することにより、従来の対訳辞書や翻訳メモリとは異なるアプローチでの翻訳支援を実現することである。

以下に本論文の構成を示す。まず 2 章において Bilingual KWIC の概要について述べ、3 章においてその特徴を紹介する。4 章において、Bilingual KWIC の技術的詳細を、5 章においてその開発過程をそれぞれ述べる。6 章ではユーザによる Bilingual KWIC の評価について述べ、7 章では類似するシステムとの比較を行う。8 章は本論文のまとめである。

¹ <http://www.japaneselawtranslation.go.jp/>

² <http://nutriad.provost.nagoya-u.ac.jp/>

2 Bilingual KWIC の概要

Bilingual KWIC の概観を図 1 に示す。これは Bilingual KWIC が「文脈検索」という名前で採用されている JLT 版での画面である。なお、Bilingual KWIC 自体は JLT に先立って開発されたものであり、以降の説明は、特に断わりがない限り、JLT 版以外の Bilingual KWIC に共通するものである。

左上のキーワード入力欄にキーワードを入力し、その横の [検索] ボタンを押すと、左側に原言語、右側に対象言語で対応付けられた対訳文を表示する。その際、原言語ではキーワードを中心に、また、対象言語では自動的に推定したその対訳表現を中心に、それぞれ KWIC 形式で表示する。また、注目したい文をマウスでクリックすると、その文全体が下側に表示される。その際、コーパスが複数の文書から構成される場合には、注目文の出典である文書名を表示できる。

右上の対訳表現入力欄には Bilingual KWIC が推定した対訳表現が表示されるが、それが間違っていたときには、ユーザが自分で適切な対訳表現をここに入力し、[訳語再検索] ボタンを押

図 1 Bilingual KWIC の概観

図 2 “negotiation” に対する実行結果

すことにより再表示できる。それに加えて、この欄右の [▼] ボタンを押すと、Bilingual KWIC が推定した他の対訳表現が表示され、ユーザは別の対訳表現を選択することができる。なお、キーワードおよび対訳表現ともにコーパス中における出現回数がそれらのすぐ横に表示され、対訳表現選定の一助となっている。

また、KWIC 表示欄の上部にある [並び替え] と表示された部分をクリックすることにより、出力結果をソートすることが可能である。原言語欄もしくは対象言語欄の中心にある表現の左側・右側でそれぞれソートすることができる。これにより対訳表現や用例の比較が簡単に行える。図 1 では、キーワード「専用利用権」の右側に続く語を基準にソートされている。

なお、図 1 では、原言語が日本語、対象言語が英語となっているが、キーワード入力欄に英単語を入力すれば、図 2 のように原言語を英語、対象言語を日本語として対訳表現の自動抽出が行える。またキーワード・対訳表現ともに複合語を含め、任意の文字列を入力することができる。

3 Bilingual KWIC の特徴

Bilingual KWIC は次のようなことが可能であるという特徴を持つ。

- (1) 対訳表現の抽出における誤りの訂正
- (2) 派生表現の獲得
- (3) 訳し分けに関する知見の獲得
- (4) 対訳辞書との併用
- (5) 対訳表現の指定
- (6) 他の言語への応用

以下、これらの特徴を詳しく述べる。

3.1 対訳表現の抽出における誤りの訂正

Bilingual KWIC では、自動対訳抽出における誤りをユーザが簡単に修正できる。図 1 の例では、JLT 版で公開されている日本法令の日英対訳コーパスを「専用利用権」をキーワードとして検索している。正しい対訳は、“exclusive exploitation right”であるが、自動対訳抽出の結果は“exclusive exploitation”となっている。しかし、図 1 の KWIC 形式で表示された英語文において“exclusive exploitation”の右側の文脈を見れば、“exclusive exploitation right”が対訳として正解であることが直観的に理解できる。

3.2 派生表現の獲得

Bilingual KWIC では、派生表現とその対訳を容易に獲得できる。図 1 の例では、「専用利用権」“exclusive exploitation right”の下に続いて表示される用例から「専用利用権者」という派生語と、その対訳“holder of an exclusive exploitation right”を得ることができる。

このような特徴は対訳表現の抽出を支援するときには有用であるが、対訳表現に加えて、実際の用例も表示されることから、Bilingual KWIC は翻訳支援に対しても有用である。

3.3 訳し分けに関する知見の獲得

翻訳支援システムとして捉えたとき、Bilingual KWIC は訳し分けに関する知見が容易に獲得できる点が優れている。図 2 の例では、“negotiation”の対訳として、「譲渡」と「交渉」が表示されている。そのような場合には、前後の文脈から、対訳語がどのように使い分けられているかを比較することが容易であり、この例では“between”が後続する場合は「交渉」と翻訳するのが適当であるといった知識を得ることができる。

3.4 対訳辞書との併用

Bilingual KWIC では、あらかじめ対訳辞書が用意されていれば、それを組み込んで併用することも可能である。辞書に対訳表現が登録されている場合は、その対訳を優先的に表示し、その後、自動的に推定した対訳表現を含む文を表示する。図 2 の例では、「譲渡」がこれに該当し、辞書に登録されている対訳であることを示すために緑色で表示される。その下に続く「交渉」は自動推定された対訳であり、辞書に登録されているものと区別するために青色で表示される。

JLT においては、「法令用語日英標準対訳辞書」が公開されており、JLT 版 Bilingual KWIC に組み込まれている。

3.5 対訳表現の指定

Bilingual KWIC では、対訳表現の抽出に失敗した場合や、特定の対訳表現に注目したい場合

図 3 「検証」の訳語として“review”を指定した結果

は、対訳表現入力欄にそれを入力することにより、対訳を指定できる。図 3 の例では、「検証」の訳語として少数ながら出現した“review”を入力することにより、その用例を表示している。なお、ユーザが指定した対訳表現は赤色で表示され、辞書にある対訳よりもさらに優先して表示される。

3.6 他の言語への応用

また Bilingual KWIC は、後述するように形態素解析を利用せず、文字レベルの情報だけを利用しているため、様々な言語対での利用が可能である。図 4 は、ベトナム語と英語の対訳コーパスを利用した例である³。なお、この例のように、言語の種類の判定が入力文字列からでは容易でない言語対で利用する場合は、入力言語を切り替えるトグル・ボタンをキーワード入力欄の左に用意している。

図 4 ベトナム語–英語コーパスへの適用

³ これは JLT 版ではなく、別のウェブ・サーバ上で実行した例である。

4 Bilingual KWIC の技術的詳細

本章では、Bilingual KWIC の技術的な詳細について述べる。まず、対訳表現の自動抽出手法について述べたあと、どのような文字列を対訳ペアの単位とするかについて述べる。

4.1 対訳表現の自動抽出

本節では、Bilingual KWIC における対訳表現の自動抽出の手法について述べる。

4.1.1 Dice 係数の利用

対訳コーパスから対訳表現を自動的に抽出する手法については、これまでも様々な手法が提案されている。

基本的な手法は、対訳コーパスから統計的に得られる情報をもとに類似度を計算するものである。すなわち、対訳コーパスの中から対訳表現の候補を求め、入力語との類似度を計算して、もっとも類似したものを対訳表現として抽出する。類似度としては、Dice 係数・相互情報量・ ϕ^2 統計量・対数尤度比などが知られているが、これらに関しては文献 (Matsumoto and Utsuro 2000) が詳しい。また、類似度として翻訳確率を利用する IBM モデル (Brown, Pietra, Pietra, and Mercer 1993) とそれを HMM に基づいて実装した GIZA++ (Och and Ney 2003) も広く利用されている。IBM モデルの場合、与えられた入力語の対訳表現を求めるのではなく、原言語文中の各単語と対象言語文中の各単語の間に対応を付ける。さらに対応付けを単語の連続として表現されるフレーズに拡張したものとして、統計的機械翻訳システム Moses (Koehn, Hoang, Birch, Callison-Burch, Federico, Bertoldi, Cowan, Shen, Moran, Zens, Dyer, Bojar, Constantin, and Herbst 2007) が出力するフレーズ・テーブルも利用されている。GIZA++ や Moses は、出現回数がある一定以上の単語やフレーズに対しては高い精度で対応を付けることが可能である。その他、統計情報に加えて、既存の対訳辞書など、他の言語情報を利用する手法も提案されている (熊野, 平川 1994; 出羽 2004)。

Bilingual KWIC においては、より多くの言語対に適用できるようにするため、言語情報はできるだけ利用せず、統計情報だけから対訳表現を求めることとした。この場合、高い精度をもつ GIZA++ や Moses の結果を利用することも考えられるが、GIZA++ は単語、Moses はフレーズを単位として対応を付けるため、単語の一部などのような単位にそぐわない表現に対しては対訳を推定することができない。Bilingual KWIC では、ユーザがキーワードを入力することを前提としており、特に日本語のように 1 単語の定義が曖昧な言語では、入力キーワードがシステムが想定する単位に合致しない場合が発生しやすいと考えられる。

そこで、どのようなキーワードに対しても対訳表現を計算できるよう、Bilingual KWIC では類似度を計算する手法を採用した。類似度にも複数の候補があるが、文献 (北村, 松本 1997)

において、候補間の類似度を計算する手法として相互情報量と Dice 係数を比較し、Dice 係数の方が高い精度を出すことが示されている。これを参考にし、Bilingual KWIC でも以下に示す Dice 係数を類似度として採用した。

$$Dice(x, y) = \frac{2 \times freq(x, y)}{freq(x) + freq(y)} \quad (1)$$

ここで $freq(x)$ と $freq(y)$ は、入力キーワード x および対訳表現候補 y がそれぞれ原言語コーパスおよび対象言語コーパス中に出現する回数であり、 $freq(x, y)$ は、対応付けられた文に x と y が同時に出現する回数である。よって、 $0 \leq Dice(x, y) \leq 1$ となる。実際に Bilingual KWIC で使用する場合は、入力キーワード x を固定した上で、 $Dice(x, y)$ が最大となる y を探すことになることから、以下の式を用いる。

$$\hat{y} = \arg \max_y \frac{2 \times freq(x, y)}{freq(x) + freq(y)} \quad (2)$$

実装においては、まずキーワード x が出現した文に対応する対訳文を集めて探索範囲とし、その中に出現するあらゆる候補 y について上記の式 (1) を計算し、最大となるものを求めている。

4.1.2 再帰的な対訳表現の抽出

上述の式 (2) を使用した場合、 $\hat{y}$ は一つしか求められない。しかし、実際には同じ入力キーワードが複数の対訳表現をもつことがある。そこで、Bilingual KWIC では、下記に示す方法で複数の対訳表現を再帰的に抽出する。

まず、最初の $\hat{y}$ が求まった場合、探索範囲から $\hat{y}$ を含んだ文を削除する。そして残った文を新たな探索範囲とし、式 (2) を再度計算することにより、異なる対訳表現を求める。探索範囲に含まれる文数や、 $\hat{y}$ に対する Dice 係数の値が閾値以下になった場合は計算を終了し、そうでない場合はさらに対訳表現を再帰的に求める。

単純に Dice 係数の値が大きな順に対訳候補とすると、最初に求めた表現の部分文字列などが含まれる場合があるが、既に抽出した対訳表現が出現しない文から対訳候補を求めることにより、最初の候補とは別の対訳表現を抽出できる。

以上の方法により、図 2 のように入力キーワードが複数の対訳表現をもつ場合に、それぞれを抽出できる。

4.2 文字レベルの情報のみの利用

GIZA++ や文献 (北村, 松本 1997) を含め、先行研究では日本語・英語とも形態素解析を利用するものが多いが、Bilingual KWIC では、形態素解析を利用せずに文字レベルの情報だけを

用いている。ここで文字レベルの情報とは、日本語の平仮名は対訳表現に含めない⁴、英語の単語は空白で区切られる、といった情報である。具体的には、日本語は文字 N グラム、英語は単語 N グラムを用いて、ある程度の長さ N をもつ対訳表現の候補を求めている。なお、 N の最大値は言語ごとに指定できる。

ところで、形態素解析を利用する利点として、対訳表現抽出の精度向上が期待できる点が挙げられる。特に動詞のように活用する語や、英語名詞の複数形などは、形態素解析を利用しないと、変化形が別の語として認識されてしまう。

しかし、形態素解析の誤りは対訳抽出に影響を与える、他の言語に応用する場合はその言語に対応した形態素解析システムが必要といった問題もある。さらに、形態素解析システムの辞書にない単位では利用できないという問題がある。

そうした点を考慮し、さらに対訳表現抽出の誤りを容易に修正できることから、Bilingual KWIC では形態素解析を利用しないこととした。それにより、語の一部だけをキーワードとして入力するなど、柔軟な入力も可能になった。ただし、単語の途中からを候補対象とすると、精度や速度の点で問題があるため、接頭語として含まれている場合だけを数え上げている。例えば、“search” の出現回数を数えるときには “searches”, “searching”, “searched”, “searcher” なども含めて数えている。これにより、動詞の活用形や名詞の複数形が規則変化する語については、形態素解析の利用なしでも、ある程度対処できている。

5 Bilingual KWIC の開発

本章では、Bilingual KWIC の開発について、その段階にそって述べる。

5.1 プロトタイプ版の開発

プロトタイプ版となる最初の Bilingual KWIC の実装は、2003 年に筆者が独自に行った。PC 上でスタンドアロンで動作するように設計し、使用した言語は Ruby、GUI 作成のために Tk のライブラリを利用した。Ruby はスクリプト言語であり、プログラム開発が容易である一方、実行速度が遅いという欠点がある。Bilingual KWIC の実装においては、Dice 係数の計算のために文字列の出現回数を高速に数え上げる必要がある。そこで、文字列検索の高速化のために、Suffix Array(山下 2000) のライブラリである sary⁵ を使用した。なお、今日では当然であるが、文字コードには UTF-8 を採用した。これにより、図 4 の例のように、各種言語に対応した。

⁴ オプションで含めることも可能である。

⁵ <http://sary.sourceforge.net/>

5.2 対訳辞書作成支援版の開発

プロトタイプ版 Bilingual KWIC は, JLT(外山 他 2012) で公開される「法令用語日英標準対訳辞書」の構築において利用された(外山, 小川 2008). その際には, この標準対訳辞書に収録する対訳ペアの候補を収集することが必要になるため, Bilingual KWIC 上から対訳ペアを候補として登録できるようにした. 具体的には, 登録したい単語を指定して右クリックすると図 5 に示すポップ・ウィンドウが表示されるようにした. このウィンドウ上で, 必要に応じてデータを修正・追加して, 登録できるようにした.

実際に, このシステムがインストールされたノート PC を作業担当者に配布することにより, 標準対訳辞書に収録する対訳ペアの候補が収集された. この時点では, Bilingual KWIC が利用する対訳コーパスの大きさは 39,560 文であったが, ノート PC 上で問題なく動作していた.

なお, この対訳ペア登録機能は, 対訳辞書の構築には有用であるが, 翻訳支援の目的とは異なるため, 次節のウェブ対応版では採用しなかった.

5.3 ウェブ対応版の開発

プロトタイプ版はスタンドアロンな PC 上で起動するが, あらかじめ Ruby の処理系などを用意する必要がある, インストールは簡単ではなかった. また, 利用者が各自で対訳コーパスを用意する必要がある, 広く使用してもらうことができなかった.

そこで多くのユーザに利用してもらうために, ウェブ・サーバ上で動作可能なバージョンを開発した. 基本的なエンジンはプロトタイプ版と同じであるが, ウェブ・ブラウザを利用する

図 5 対訳ペア登録用ウィンドウ

ための GUI 部分の開発は業者に委託した。この時点で、ユーザ・インターフェイスも含めて、Bilingual KWIC は一通り完成した。

なお、Bilingual KWIC の完成後、日本法令外国語訳データベースシステム (JLT) の開発が始まったが、Bilingual KWIC は翻訳支援における有用性も認められ、2009 年の JLT 開設当初から「文脈検索」の名前で採用されている。その際には、より多くのウェブ・ブラウザに対応させるなどの改良を施した。これにより多くの一般ユーザに Bilingual KWIC を利用してもらえるようになった。

5.4 高速化

最初のプロトタイプ版では、対訳コーパスとして数万文程度のサイズを想定していた。しかし、JLT 版では新しい対訳法令が次々に追加されるため、コーパスのサイズが 10 万文を超える段階において、実行速度の点で問題が発生した。特に入力キーワードの出現回数が 1 万を超えるような場合は、結果が表示されるまでに数十秒かかることもあった。

この問題には 2010 年から取り組み、以下の二つの方法で対処した。この手法は JLT 版には適用していないが、1 章で述べた NUTRIAD 版において適用している。

5.4.1 対訳表現抽出の高速化

実行速度が遅い原因の一つは、コーパスが大きくなると対訳表現抽出に時間が掛かるためである。式 (2) においては、 $\hat{y}$ を求めるために、あらゆる対訳表現候補 y について $freq(y)$ と $freq(x, y)$ の計算が必要となる。 y は、上限（デフォルトは日本語で 6 文字、英語で 4 単語）までのあらゆる N グラムを候補とするため、入力キーワード x が出現する文数が大きくなると、 $freq(y)$ と $freq(x, y)$ の計算回数がそれだけ多くなる。このうち $freq(y)$ の計算は、コーパスの並び替えとインデックス化を事前に行う Suffix Array を用いることにより、高速に実行できるため問題ない。しかし、 $freq(x, y)$ に関しては、 y の出現回数を数える探索範囲が x に依存して変化する。そのため、探索範囲を事前にインデックス化することが不可能であり、Suffix Array を用いた高速化ができない。そこでプロトタイプ版および JLT 版では、探索範囲内を順次 N グラムに分割し、その出現回数を数え上げていた。

この $freq(x, y)$ の計算を高速化するため、コーパスの各文をあらかじめ一定サイズ以下の N グラムに分割したデータを別に保持し、それを数え上げることとした。すなわち、デフォルトでは日本語は 8 文字以下、英語は 10 単語以下の N グラム単位であらかじめ分割したデータを保持しておく。また実装においては、テキストデータ自体へのアクセスを高速化するために Tokyo Cabinet⁶を導入した。

⁶ <http://fallabs.com/tokyocabinet/>

5.4.2 表示の高速化

実際の利用の上では、表示速度においてもボトルネックがあった。入力キーワード x が出現する文が多くなると、ブラウザ側に負担が掛かり、表示が遅くなっていた。そのため、JLT 版においては、デフォルトで 100 文を超える分は表示しないこととし、オプションとして表示できる文数の上限の設定を 200, 400, 800 と変更できるようにした。このため、入力キーワードを含むすべての対訳文を表示することができない場合があるが、通常の利用では、800 文を表示できれば十分に比較ができると考えた。また、入力キーワードに文字列を付加したり、対訳表現を指定することにより、希望する対訳文を表示することができる。

しかし、JLT 版においては、別の問題があった。JLT 版においては、表示結果のソートをブラウザ側で実現していた。しかし、これもブラウザに負担を掛け、表示が遅くなる結果となっていた。よってソート自体をサーバで実行し、その結果をブラウザ側に再送して再表示することにより、高速化を実現した。

5.4.3 高速化の効果

上述した二つの高速化手法の効果を測定するために、高速化を適用していない JLT 版と、適用した NUTRIAD 版とにおける表示速度を比較した。具体的には、出現回数が異なるキーワード 50 語を選び、そのキーワードを入力してから結果が表示されるまでの時間を測定した。時間の計測には、ウェブ・ブラウザを利用してウェブ・アプリケーションをテストするツールである Selenium WebDriver⁷を Ruby から使用した。表示する文数の上限はデフォルト値である 100 とした。また、組み込まれている対訳辞書を併用した場合、対訳表現の自動推定の一部が省略されて訳語推定の時間が異なってくるため、併用しない設定で測定した。その結果を図 6 に示す。

この結果から分かるように、結果が表示されるまでの時間は出現回数が増えるに従って増加する傾向にあるが、単純に比例する訳ではない。これは再帰的な対訳表現の抽出が主な原因と考えられる。4.1.2 節で述べたように、Bilingual KWIC では、最初の対訳表現を求めた後、それが含まれない残りの文から再帰的に対訳表現を求める。そのため、多くの対訳表現をもつキーワードほど、すべての対訳表現を抽出するまでの時間が長くなる。

JLT 版と NUTRIAD 版では、サーバの性能や登録されているコーパスの量・内容、入力されるキーワードが異なるため、単純な数値の比較はできないが、高速化を施した NUTRIAD 版の方がキーワードの出現回数が増えた場合でも、実行時間の増加が緩やかであり、高速化手法が有効であったことが分かる。

⁷ <http://www.seleniumhq.org/projects/webdriver/>

図 6 出力時間の比較

6 ユーザによる評価

本章では、実際に利用したユーザに対するアンケートに基づく Bilingual KWIC の評価について述べる。

6.1 アンケートの対象と内容

法学部の講義の一環として法令翻訳の課題が実施されており、受講生は Bilingual KWIC を利用して法律文を翻訳している。今回は、この講義の受講生に Bilingual KWIC の評価を依頼した。

受講生は、まず初回の講義において、使用する道具に関する説明なしで与えられた法律文を翻訳した。それから、次の回の講義において JLT 版の Bilingual KWIC である「文脈検索」の説明を受けた後、改めて別の法律文を翻訳した。なお、説明の内容は、キーワードを入力して結果が表示される例を示すという単純なものであり、一般のユーザがウェブサイトに辿り着いてキーワード入力を実行し、その動作を理解する場合と同程度と想定した。

Bilingual KWIC を使用しない場合と使用した場合の両方の翻訳を試みることにより、Bilingual KWIC が翻訳に有用であったかを評価してもらった。評価項目は、表示の見やすさ（視認性）・

図 7 ユーザによる評価結果

訳語の推定精度（推定精度）・表示速度・役に立ったか（有用性）に関する満足度であり、それぞれ5段階で評価してもらった。

6.2 評価結果

受講生 49 人に対するアンケートの集計結果を図 7 に示す。視認性と訳語の推定精度に関しては、約 5 割が満足している一方で、表示速度に関しては 4 割以上が不満を感じている。JLT 版の Bilingual KWIC には、5.4 節で述べた高速化が適用されていないため、その点も低い評価に繋がっていると考えられる。しかし、有用性に関しては 8 割以上が肯定的な評価であり、Bilingual KWIC が翻訳支援として役に立つことが示された。

7 関連研究

コーパスに対して検索や分析を行うツールはコンコーダンスと呼ばれるが、これを 2 言語コーパスに拡張したものはバイリンガル・コンコーダンスもしくはパラレル・コンコーダンスと呼ばれる。Bilingual KWIC はバイリンガル・コンコーダンスの一種であると言える。本章では、いくつかのバイリンガル・コンコーダンスについて紹介し、Bilingual KWIC との違いについて述べる。

文献(内山, 井佐原 2003)における翻訳メモリの利用方法はバイリンガル・コンコーダンスといえる。ここでは、与えられたキーワードの日本語コーパスにおける出現を KWIC 形式で表示し、コーパス中においてキーワードと良く共起した日本語単語と英語単語を提示する。ユーザが対訳候補となる英語単語を選ぶと、それに基づいた絞り込み検索を行い、対訳文のペアを別

ウィンドウに表示する。ただし、別ウィンドウに表示される対訳文のペアでは、キーワードと対訳候補の部分がそれぞれ下線で明示されるが、中心に揃えて表示される訳ではない。Bilingual KWIC では、最初に対訳候補を自動的に決定する点、任意の対訳表現を指定できる点、原言語だけでなく対象言語も同時に KWIC 形式で表示する点が異なる。

TransSearch⁸ (Macklovitch, Simard, and Langlais 2000) はオンラインのバイリンガル・コンコーダンスであり、原言語文と対象言語文を左右に並べて表示する。当初は原言語文中の入力キーワードがハイライトされるだけであり、対訳語はユーザが推測する必要があった。文献 (Bourdaillet, Huet, Langlais, and Lapalme 2010) による改良により、対訳語の自動推定機能が追加され、対訳語はハイライトされるようになったが、KWIC 形式の表示は導入されていない。一方で、単数形と複数形などのように類似した訳語をまとめるなど、Bilingual KWIC にはない機能を備えている。

Linear B (Callison-burch and Bannard 2005) はパラレル・コーパスを翻訳メモリとして捉え、検索できる翻訳メモリというコンセプトに基づくバイリンガル・コンコーダンスである。原言語文と対象言語文が上下交互に表示される形式であり、KWIC 形式は導入されていない。対訳語は自動推定されるが、同じ対訳語を含む文が多数ある場合、その一部だけを表示し、複数の対訳語が同じ画面に表示されるようになっている点が Bilingual KWIC とは異なる。

なお、TransSearch および Linear B の自動推定は GIZA++ や Moses などでも利用される単語やフレーズの対応付け技術を応用したものであり、キーワードが入力される前にあらかじめ推定している点が Bilingual KWIC と異なる。よって、両者では対訳自動推定の精度をいかに向上させるかが重視されており、特に文献 (Bourdaillet et al. 2010) では様々な自動推定の方法が比較・検討されている。一方、Bilingual KWIC では表示方法を工夫することにより、ユーザが自動推定の誤りを訂正しやすくするというアプローチで、自動推定の誤りに対処している。

原言語文と対象言語文の両者を KWIC 形式で表示するバイリンガル・コンコーダンスとしては ParaConc⁹ (Barlow 2004) が挙げられる。ただし、その KWIC 表示は半自動というべきものである。ユーザがキーワードを入力すると、原言語文は上ウィンドウにおいて KWIC 形式で表示されるが、対象言語は下ウィンドウに通常の形式で表示される。その際、対訳語の候補が示されており、それをクリックすることにより KWIC 形式での表示に変化する。ただし、原言語文と対象言語文が上下のウィンドウで完全に分割されており、例えば上ウィンドウの 3 文目に対応する文は下ウィンドウの 3 文目に表示されるといった具合であり、左右に並べて表示する Bilingual KWIC と比べた場合、原言語文と対象言語文の対応は分かりにくい。なお、対訳語候補の選出方法は出現頻度に基づくものであり、Dice 係数と類似したものと考えられるが、具体

⁸ <http://www.tsrali.com/>

⁹ <http://paraconc.com/>

的な計算式が掲載されていないため、詳細は不明である。

8 まとめ

本論文では、対訳表現抽出を可視化することで翻訳を支援する Bilingual KWIC の開発について述べた。Bilingual KWIC は、任意の入力キーワードに対して対訳表現を自動抽出し、パラレル・コーパス中での用例と一緒に提示することにより、ユーザの翻訳作業を支援する。

本システムは、既に述べた通り、法務省・日本法令外国語訳データベースシステム (JLT) および名古屋大学・学内情報翻訳データベース NUTRIAD において採用されている。

現在、JLT では 36 万文 150 MB 以上からなる対訳コーパスを用いて Bilingual KWIC を運用している。出現回数の少ない入力キーワードに対しては問題なく動作しているが、5.4 節で述べた高速化が適用されていないため、出現回数が多いキーワード、特に出現回数が 1 万回を超えるような場合には結果が表示されるまでに数十秒かかることがある。また、高速化が適用された NUTRIAD 上の実装においても、出現回数が極端に多いキーワードに対しては応答に時間がかかっており、Bilingual KWIC の一層の高速化が求められている。それに対しては、出現回数や入力頻度の多いキーワードに対する計算結果をキャッシュしておくなどの対応を検討している。さらには、Moses などを利用してフレーズ・テーブルをあらかじめ計算しておき、入力キーワードがフレーズ・テーブルにある場合はその結果を、そうでない場合は Dice 係数によりその場で計算するなどのハイブリッドな手法を導入することにより、高速化と高精度化を同時に実現する方法も検討している。

謝 辞

Bilingual KWIC の開発にあたっては、ウェブ版インターフェイスの開発、高速化などにおいて株式会社リーガルアストレイに協力していただいた。ユーザによる評価実験においては、名古屋大学大学院法学研究科附属法情報研究センターの中村誠特任准教授と佐野智也特任講師に協力していただいた。

参考文献

- Barlow, M. (2004). “Parallel Concordancing and Translation.” In *Proceedings of ASLIB Translating and the Computer*, Vol. 26.
- Bourdaillet, J., Huet, S., Langlais, P., and Lapalme, G. (2010). “TransSearch: from a Bilingual Concordancer to a Translation Finder.” *Machine Translation*, **24** (3–4), pp. 241–271.

- Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). “The Mathematics of Statistical Machine Translation: Parameter Estimation.” *Computational Linguistics*, **19** (2), pp. 263–311.
- Callison-burch, C. and Bannard, C. (2005). “A Compact Data Structure for Searchable Translation Memories.” In *10th EAMT Conference: Practical Applications of Machine Translation*, pp. 59–65.
- 福田薫, 外山勝彦, 野田昭彦 (2013). 学内情報翻訳データベースの構築と運用. 大学 ICT 推進協議会 2013 年次大会論文集, pp. 146–152.
- 出羽達也 (2004). 対訳文書から自動抽出した用語対訳による機械翻訳の訳語精度向上. 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, **87** (6), pp. 1244–1251.
- 北村美穂子, 松本裕治 (1997). 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, **38** (4), pp. 727–736.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). “Moses: Open Source Toolkit for Statistical Machine Translation.” In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL ’07*, pp. 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 熊野明, 平川秀樹 (1994). 対訳文書からの機械翻訳専門用語辞書作成. 情報処理学会論文誌, **35** (11), pp. 2283–2290.
- Luhn, H. P. (1960). “KeyWord-In-Context Index for Technical Literature (KWIC Index).” *American Documentation*, **11** (4), pp. 288–295.
- Macklovitch, E., Simard, M., and Langlais, P. (2000). “TransSearch: A Free Translation Memory on the World Wide Web.” In *2nd International Conference on Language Resources and Evaluation (LREC)*, pp. 1201–1208.
- Matsumoto, Y. and Utsuro, T. (2000). “Lexical Knowledge Acquisition.” In Robert, D., Hermann, M., and Harold, S. (Eds.), *Handbook of Natural Language Processing*, pp. 563–610. Marcel Dekker.
- Och, F. J. and Ney, H. (2003). “A Systematic Comparison of Various Statistical Alignment Models.” *Computational Linguistics*, **29** (1), pp. 19–51.
- 外山勝彦, 小川泰弘 (2008). 自然言語処理の応用に基づく法令外国語訳支援. 人工知能学会誌, **23** (4), pp. 521–528.
- 外山勝彦, 齋藤大地, 関根康弘, 小川泰弘, 角田篤泰, 木村垂穂, 松浦好治 (2012). 日本法令外国語訳データベースシステムの設計と開発. 情報ネットワーク・ローレビュー, **11**, pp. 33–53.
- 内山将夫, 井佐原均 (2003). 日英新聞記事対応付けデータを用いた翻訳メモリと言語横断検索.

情報処理学会全国大会講演論文集, 65-5 巻, pp. 355-358.

山下達雄 (2000). 用語解説「Suffix Array」. 人工知能学会誌, **15** (6), p. 1142.

略歴

小川 泰弘：1995 年名古屋大学工学部情報工学科卒業。2000 年同大学院工学研究科情報工学専攻博士課程後期課程修了。同工学研究科助手，同情報科学研究科助教を経て，2012 年名古屋大学情報基盤センター准教授（同大学院情報科学研究科兼任）。現在に至る。博士（工学）。自然言語処理および法律情報処理に関する研究に従事。言語処理学会，情報処理学会，電子情報通信学会各会員。

外山 勝彦：1984 年名古屋大学工学部電気学科卒業。1989 年同大学院工学研究科情報工学専攻博士課程満了。同工学部助手，中京大学情報科学部講師，同助教授，名古屋大学大学院工学研究科助教授，同情報科学研究科助教授を経て，2013 年名古屋大学情報基盤センター教授（同大学院情報科学研究科兼任）。現在に至る。工学博士。論理に基づく知識処理，自然言語処理に関する研究に従事。近年は法制執務支援や法令翻訳支援に関心を持つ。言語処理学会，電子情報通信学会，情報処理学会，人工知能学会，日本認知科学会，ACL 各会員。

(2016 年 5 月 20 日 受付)

(2016 年 8 月 3 日 再受付)

(2016 年 9 月 15 日 採録)