

## <論文>

# ソフトウェア UI 英語のレジスター分析: 目標テキスト生成能力の向上に向けて

## Register Analysis of Software UI English: For Improving TT Generation Competence

西野 竜太郎 野原 佳代子

(東京工業大学)

### Abstract

Localization includes translation of digital content such as software, video games, or websites. In addition to being the language spoken in the United States, which is an important sales market, English often serves as a pivot language in localization. In other words, the ability to translate into English is indispensable for many localizers. In this paper, we study English used in software user interfaces (UIs) with the register analysis method so that results can be utilized in translator education to improve the target text generation competence. The results show that, compared with other registers, UI English frequently uses modals (*can*), 2nd person pronouns, and demonstrative determiners (*this*). It also contains more imperatives, more sentence fragments, and placeholders that appear only in software. These linguistic characteristics come from the situational characteristics such as shared contexts between a user and software.

### 1. はじめに

ローカリゼーションとは、製品を特定の言語や文化に適合させるプロセスで、そこには翻訳が含まれる (Globalization and Localization Association, n.d.)。例えば輸出車のハンドル位置変更も広い意味でローカリゼーションではあるが、言語間翻訳が関係する場合はソフトウェアやウェブサイトといったデジタル情報が対象となる。翻訳方略としては、目標志向 (*target-oriented*) が採用される (Jiménez-Crespo, 2013; 山田, 2007) 点の特徴である。「適合」がローカリゼーションであるならば、翻訳方略が目標志向となることに不思議はない。また別の特徴として「ピボット言語」を用いて翻訳されることがある点が挙げられる (山田, 2013)。例えば日本企業が強みを持つビデオゲーム分野では、まず日本語からアメリカ英語に翻訳し、そこから FIGS (仏伊独西) およびイギリス英語に展開する (O'Hagan & Mangiron, 2013)。同様にソフトウェア分野でも、アメリカ国外の企業はまず英語で開発し、そこから多言語翻訳することがある (Esselink, 2000)。これは、アメリカが主要な販売マーケットである、英語を扱う翻訳者が豊富であるといった理由でアメリカ英語がピボットとなっているのではないかと想像できる。

このようにローカリゼーションにおいては、起点言語が何であれ、英語化 (英訳) のニーズが存在する。アメリカ市場での販売を狙って英語を目標言語とする場合はもちろん、上記のピボット言語のような使われ方もする。インターネットが世界的に普及し、スマートフォンのアプリやゲーム、ウェブサイト、電子書籍などが国境を超えて容易に提供されるようになれば、英語化やピボット言語によ

る多言語化の機会はさらに増えるであろう。

## 1.1 目的

そこで本研究では、ローカリゼーションにおける「目標言語としての英語」について言語的な特徴を明らかにする。ローカリゼーションはウェブ、ビデオゲーム、ソフトウェアなどに分類される (Jiménez-Crespo, 2013) が、本研究ではソフトウェアを対象とする。特に注目するのはユーザー・インターフェイス (user interface、UI) である。目標言語としてのソフトウェア UI 英語の特徴を明らかにし、それをローカリゼーション教育や学習の場で扱うことで、翻訳者の能力向上に役立つはずだ。実際、翻訳能力とは適切な目標テキストを生成して選択できるスキルだという指摘 (Pym, 2003) がある。またローカリゼーションの能力モデルで、バイリンガル・サブコンピテンスの 1 つとして「目標言語における高度なテクニカル・ライティング・スキル」が挙げられている (Jiménez-Crespo, 2013)。

目標言語における言語的特徴として注目するのは「レジスター」(言語使用域) である。レジスターとは、「ある状況において適切であると言語使用者が考える言語の変種」(Baker, 2011) のことである。異質化の場合のように意図的に起点テキストの特徴を出す場合を除き、翻訳者は通常、想定読者が持つレジスター的な期待に合うよう翻訳する (Baker, 2011)。そのため、翻訳者のスキルとしてレジスターを使い分けられることが重要となる。レジスターに合わない訳文というのは、意味や文法の解釈自体に誤りがあるわけではなく、ある状況で適切でないということである。以下にソフトウェア UI における例を挙げてみる。

### 1.1.1 レジスターに合う翻訳と合わない翻訳の例

交流サイトの「Facebook」(Facebook, n.d.) ではボランティアのユーザーが UI の翻訳をしている (クラウドソーシング翻訳)。あるユーザーの訳に対して他のユーザーが投票し、より多くの票を得た訳文が採用される。こうして採用された訳文はレジスターに合っているものだと想像できる。この Facebook を例としてレジスターを説明するが、Facebook の起点言語は英語のため英日の事例を挙げる。

1 つめは、写真アルバムの脇に付けられた「Empty」という説明だ。訳文候補として「空」、「なし」、「写真がありません」などが挙げられているが、最終的には「写真がありません」が採用されている (2014-07-30 時点)。「空」や「なし」が翻訳として誤っているのではなく、「ありません」と丁寧な表現にした方がこの状況では適切だということだろう。2 つめは、メモ書きを他人に公開したときに表示される「Your note was published successfully.」というメッセージである。訳文候補として「あなたのノートが公開されました。」や「ノートが公開されました。」が挙げられているが、実際に採用されているのは後者である。この状況における日本語のレジスターでは「あなたの」と明示しない方が適切なのだろう。

このように、翻訳では目標言語のレジスターを意識することが重要で、当該レジスターの特徴を知っておけば、より適切な翻訳が可能となるはずだ。そのような翻訳ができれば、読者やユーザーが見慣れた表現であるため読みやすさや使いやすさが向上したり、言葉遣いに注意を払った質の高い製品だという印象を与えられたりといった効果が期待できる。

つまり本論文では、特にレジスターという視点から UI 英語の言語的特徴を明らかにすることで、ソフトウェア・ローカリゼーションに関わる翻訳者や学習者の目標テキスト生成能力の向上に資することを目的としている。また、翻訳研究における新しい方法として、本研究で用いるレジスター分析手法を提案することも目的としている。このような方法で既存のレジスター（コンピューター、法律、医薬など）をあらかじめ分析しておけば、その成果を翻訳教育やライティング教育で応用できるのではないだろうか。

## 1.2 関連研究

ソフトウェア UI の言葉に関する研究はコンピューター分野、とりわけヒューマン・コンピューター・インタラクション (HCI) で行われている。グラフィカル・ユーザー・インターフェイス (GUI) 以前のコマンド・ライン・インターフェイス (CLI) の時代から研究は存在し、1970 年代頃には特にエラー・メッセージが注目された (例: Shneiderman, 1979)。80年代の GUI 登場や90年代のウェブサイト普及後も同じで、ユーザビリティを高めるには良いエラー・メッセージが必要であるとされた (Nielsen, 1994)。つまり、HCI 分野における UI 言語に対する関心はエラー・メッセージが中心であった。

言語学や翻訳学の分野からのソフトウェア UI 英語に対するアプローチ、特にコーパスを用いた研究は多くない。例えば Jiménez-Crespo (2009) はスペイン語化された企業ウェブサイトについてコーパスを作成し、メニュー名などの UI の翻訳を調査している。ウェブやソフトウェアのローカリゼーションでは慣習的 (conventional) な言葉遣いが重要で、そういった慣習はコーパスを使わないと見つけられないと述べている。Nishino & Nohara (2013) はコーパスを分析し、一般英語に比べて UI 英語は非母語話者にとって難易度が高い言葉が使われているものの、語彙の種類は少ないと結論づけている。

このように、コーパスを用いて UI 英語のレジスターを分析するという先行研究はほとんど存在しない。コーパスを活用する本研究は、その点で意義があると考ええる。

### 1.2.1 さまざまなレジスター分析

本研究では「レジスター分析」を用いる。分野や研究者によって手法が異なるが、翻訳研究におけるレジスター分析は選択体系機能言語学の方法が基本となっており、フィールド、テナー、モードという3要素 (ハリデー & ハッサン, 1991) から分析する。例えば House (2006) はその3要素から ST と TT を分析して比較し、overt または covert という2種類の翻訳をすすとしている。また Hatim & Mason (1990) は、ある新聞記事について「フィールド: 報道、テナー: 公正で事実に基づく態度、モード: 読まれる目的の書き言葉」のように具体的に分析した例を挙げている。翻訳者教育を目的とした *In Other Words* (Baker, 2011) で Baker は上記の3要素の紹介に加え、あるレジスターで特有のコロケーションを知ることが翻訳者にとって重要であると述べ、例えばコンピューターの文書で英語の data という名詞は handled、extracted、processed などの動詞と結び付くといった例を挙げている。

本研究では上記の翻訳研究におけるレジスター分析手法ではなく、Biber & Conrad (2009) によるレジスター分析手法を用いる。これは、ある既存のレジスター (例えば新聞) の特徴を調べるために、別のレジスターと定量的な比較を行う方法である。翻訳研究における方法は基本的に、翻訳時に翻訳

対象テキストを初めて見た際に行うことが想定されている。つまり、初見のレジスター不明のテキストを分析し、翻訳作業に役立てるのに使われる。一方 Biber & Conrad による方法は、何らかのレジスターのテキストを分析し、言語的特徴を記述する場合に使われる。両者ともに「レジスター分析」と呼ばれているが、このような違いがある。本研究ではソフトウェア UI 英語というレジスターを分析して言語的特徴を記述したいため、Biber & Conrad の方法を採用する。

本論文では UI 英語と他のレジスターとを比較する。対象とするのは「新聞」と「ブログ」で、これらレジスターの先行研究は存在する。*Longman Grammar of Spoken and Written English* (Biber, Johansson, Leech, Conrad & Finegan, 1999) (以下 LGSWE と略) では会話、フィクション、学術文に加え、新聞のレジスターについて調査している。また *Language and the Internet* (Crystal, 2006) では電子メールなどの他に、ブログのレジスターを調査している。まず新聞を比較対象としたのは、英語話者にとって最も親しみのある書き言葉 (Biber et al., 1999) だからである。恐らくライティングの手本や模範として扱われてもいるだろうから、これとの比較で言語的特徴にどのような差があるか明らかになると、教育上有益である。またブログは比較的新しい分野で、コミュニケーションはモノローグの場合もインタラクティブの場合もある (Crystal, 2006) といった特徴を持つ。新聞とは異なる、新しい書き言葉の代表として選んだ。

## 2. 方法

前述の通り Biber & Conrad (2009) によるレジスター分析を用いる。この手法では代表的なサンプルを用いて複数のレジスターを定量的に比較する。レジスターどうしの比較により、あるレジスター (本研究では UI 英語) の特徴がよりはっきりと分かる。

### 2.1 コーパスの準備

出現頻度を数えることで、分析者の直感に頼らない定量的な比較が可能となる。本研究では少量のテキストではなく、ある程度の規模のコーパスを用いる。まず UI 英語のコーパスを構築し、続いて比較対象となる新聞およびブログの既存コーパスを取得する。

#### 2.1.1 UI 英語コーパス

コーパス化するのは、ソフトウェア UI のテキストである。現在広く用いられている UI の言語的特徴を調べたいため GUI ソフトウェアのみを収集対象とする。この際、「PC」と「モバイル」という 2 種類のカテゴリから採取しておく。画面サイズや操作方法などが原因で言語的特徴に違いが発生しないか確認したいからである。サンプル採取の際は「代表性」と「均衡性」(Sinclair, 2004) を考慮する。しかし代表性と均衡性にはどうしても恣意的な部分が残ってしまう。例えば表 1 に示すように、PC カテゴリではマイクロソフト製品 (Word, Excel など) の占める割合が大きく、均衡性に欠けるように見える。しかし、ユーザーが少なく一般的でない製品からサンプルを取得して均衡しようとした場合、今度は代表性に乏しくなる。例えば同社の「Office」は世界で 10 億人以上に使われている (Microsoft, 2013) 代表的なソフトウェアであるため、割合を高くしても非合理ではないだろう。代表

性と均衡性は「all-or-nothing」(Leech, 2007)ではなく、両方を一度に完全に達成できないため、ある程度で折り合いを付ける必要がある。また企業が UI テキスト集を提供していない、ソフトウェアからテキストを抽出できないといった理由で、現実には入手先が限定される。

そこで PC カテゴリーでは「Microsoft Word 2007 SP2」、「Microsoft Excel 2007 SP2」、「Microsoft Outlook 2007」、「Windows Media Player 10」、「Mozilla Firefox」の 5 つのソフトウェアをサンプルとした。Word、Excel、Outlook は上記の通りユーザー数が多い Office に含まれるオフィス製品、Media Player はエンターテインメント製品、Firefox はウェブ・ブラウザと、異なる用途のソフトウェアから選んだ。最初の 4 つはマイクロソフト社の製品で、UI テキストを Microsoft Language Portal (Microsoft, n.d.) から入手した。Firefox は Mozilla のウェブサイト (Mozilla, n.d.) から 2013 年 12 月 27 日時点の最新版を入手し、画面に表示されると思われるテキストを抜粋した。

続いてモバイルのカテゴリーでは「Android」、「iOS」、「Windows Phone 7.5」、「Firefox OS」の 4 つのプラットフォームをサンプルとした。現在利用されているモバイル・プラットフォームの大部分はこれらであるため、十分な代表性があると考えられる。PC カテゴリーではすべて個別ソフトウェアだったのに対し、モバイル・カテゴリーでは各プラットフォームに含まれる小規模なソフトウェア (アプリ) を対象としている。例えば Android プラットフォームには Camera、Gallery、HTML Viewer といった複数のアプリが含まれている。Android (android Git repositories, n.d.) は 2012 年 7 月 31 日時点で最新の 41 アプリ、iOS (Apple, n.d.) は 2013 年 9 月 30 日時点で最新の 152 アプリ、Firefox OS (Mozilla, n.d.) は 2013 年 12 月 27 日時点で最新の 28 アプリからテキストを取得した。Windows Phone 7.5 (Microsoft, n.d.) はアプリごとに分かれていないため全体のテキストを取得した。

収集したテキストには分析対象にならない部分が含まれ、クリーニングが必要となる。例えばプレースホルダー (%d、%s1、{0}) などや HTML タグ (<b></b>、<br>) などである。これらはソフトウェアごとに異なり、個別に対応が必要である。そこで「ユーザーに表示される文字のみを残す」という全体方針を立ててクリーニングした。ただしプレースホルダーの場合、ソフトウェア実行中に何らかの文字が代入される。例えば「ようこそ、%s さん！」のプレースホルダー「%s」にはユーザー名が入る。事前にどのような文字が入るか分からないため、プレースホルダーすべてを「variablehere」に置換し、1 ワードとして扱う。

続いてワード数をカウントした。これにはコーパス用ツールである AntConc (Anthony, 2014) を使用した。すべて小文字でカウントし、数字や記号は除外した (ただしハイフンとアンダースコアは含める)。結果は表 1 の通りである。合計 515,761 ワードで、そのうち PC は 323,222 ワード (全体の 62.7%)、モバイルは 192,539 ワード (全体の 37.3%) である。

PC カテゴリー		モバイル・カテゴリー	
Word 2007	79,783	Android	23,410
Excel 2007	108,316	iOS	116,135
Outlook 2007	101,798	Windows Phone 7.5	41,887
Media Player 10	18,712	Firefox OS	11,107
Firefox ブラウザー	14,613		
小計 : 323,222		小計 : 192,539	
<b>合計 : 515,761</b>			

表 1. UI 英語コーパスのワード数

### 2.1.2 新聞とブログのコーパス

あるレジスターの言語的な特徴は他との比較によって明確になる (Biber & Conrad, 2009) ため、比較対象となる「新聞」と「ブログ」のテキストも準備する。これは「MASC」という既存のコーパスから取得する。MASC (Manually Annotated Sub-Corpus) は Open American National Corpus のサブセットで、約 50 万ワードのコーパスである (American National Corpus Project, n.d.)。MASC は本研究で用いる新聞とブログのほか、合計 19 分野のテキストから構成されている。UI 英語と同じく AntConc でワード数をカウントしたところ、新聞は 25,417 ワード、ブログは 28,409 ワードだった。UI 英語のコーパスと比較するとワード数が少ない。ただし後に説明するが、本研究では出現頻度の比較は「1 万語あたりの回数」に標準化したり、1,000 サンプルあたりの頻度に揃えたりして行う。そのためワード数に違いがあっても比較は可能であると考えられる。

## 2.2 レジスター分析の方法

Biber & Conrad (2009) の方法では、(1) ある言葉が用いられる状況的コンテキストと (2) その言葉の言語的特徴を記述し、(3) その 2 つを機能面から解釈する。選択体系機能言語学のようなフィールド、テナー、モードという用語は使わないが、参加者は誰か、その人間関係はどうか、話し言葉か書き言葉か、コミュニケーションの目的は何かといった状況を見る。このような状況的コンテキストと言語的特徴を比較し、なぜその言語的特徴がその状況に現れたかという理由を分析するのである。またこの方法では、あるレジスターの代表的なサンプルを用い、言語的特徴をカウントし、レジスター間で定量的に比較する。

### 2.3 調査対象とする言語的特徴の候補

Biber & Conrad (2009) はレジスター分析で調査対象となる言語的特徴の候補をいくつも挙げている。例えば代名詞、助動詞、態 (能動、受動)、節や文の種類 (平叙、疑問、命令) で、こういった特徴の出現数をカウントして比較する。ただしすべての候補を調査するのは難しいため、以下のように語レベルと語を超えるレベル (節など) からいくつか選びたい。

### 2.3.1 語レベル

語レベルの調査対象を選択するために、コーパス言語学で用いられる「キーワード分析」を行う。これは2つのコーパスを比較し、一方でより多く出現する語（キーワード、特徴語）を抽出する方法である。ここではUI英語コーパスと、一般英語コーパスであるMASCとを比較し（どちらも約50万ワード）、UI英語におけるキーワードを取得する。つまりこれで調査対象候補の「あたり」を付けるのである。前述のコーパス用ツールであるAntConcにはキーワード機能があるため、それを利用する。ランク付け用の統計量として、現在コーパス言語学者に好まれている（McEnery & Hardie, 2011）とされるLLR（Log-likelihood ratio）値を用いる。この結果、表2の語がUI英語におけるトップ10のキーワードとして取得できた。

順位	LLR 値	語	順位	LLR 値	語
1	20,114.0	variablehere	6	1,928.1	document
2	3,433.3	data	7	1,921.2	settings
3	3,246.5	cannot	8	1,781.9	text
4	3,026.9	file	9	1,732.1	this
5	2,036.4	your	10	1,705.8	server

表2. 一般英語と比較した場合のUI英語におけるキーワード

このうち、1位の「variablehere」は前述の通りプレースホルダーである。著者らが仮に使っている語であり、一般英語コーパスには存在しないため1位となっている。data、file、document、settings、text、serverというキーワードはコンピューター用語であるため、UI英語に多く出現することに不思議はない。ここで注目したいのはcannot、your、thisという機能語である。各専門分野に特有の内容語より、どのレジスターにも存在する機能語の多寡にこそ、そのレジスターの特徴が現れるのではないかと考える。そこで語レベルの調査対象として、cannotを含む「法助動詞」、yourを含む「人称代名詞」、thisを含む「指示代名詞」および「指示決定詞」を選択する。これらについては「10,000ワードあたりの出現数」を計算し、レジスター間で比較する。

#### a) 法助動詞

法助動詞 (modal verb) は話し手の判断や態度を表し、can、could、may、might、must、shall、should、will、wouldの9つが中心的な法助動詞である (Biber et al., 1999:73)。本研究でもその9つを対象とする。レジスター間で判断や態度を表す表現の頻度に違いがあるのか、あるとしたらどういった種類が多いのかを分析する。前述のキーワードとして得られた「cannot」は「can」に含めてカウントする。また「we'll...」、「I'd like...」、「It won't...」といった短縮も対象とする。

#### b) 人称代名詞

一人称としてI、we、me、us、my、our、二人称としてyou、your、三人称としてhe、she、they、him、

her (目的格)、it、them、his、her (所有格)、its、their を対象とする。独立所有格 (mine など) や再帰代名詞 (myself など) は今回対象としない。人称代名詞の頻度を比較することでインタラクションの違いを観察できる。話し手 (一人称) と聞き手 (二人称) が直接コミュニケーションを取っている状況の場合、一人称と二人称の人称代名詞が頻出すると考えられる。話し手と聞き手が共に知っている第三者を話題にする状況の場合、三人称が多くなると考えられる。ソフトウェア UI の場合、ソフトウェアとユーザーのインタラクションが存在する。これがどう人称代名詞に現れるのか確認する。

#### c) 指示代名詞/指示限定詞

具体的な単語としては *this*、*that*、*these*、*those* である。*this* の場合、指示代名詞は「I like this.」、指示限定詞は「I like this apple.」という例文が挙げられる。*that* は関係詞や接続詞としての用法もあるのでカウント時にはそういったケースを除外する。ソフトウェア UI の場合、ユーザーはソフトウェアを見ながら操作する。そのため画面上 (コンテキスト) でユーザーに提示された情報やファイルなどを指示代名詞や指示限定詞を使って指し示すはずである。そういった言葉が他のレジスターと比べて多いのか少ないのかを確認したい。

### 2.3.2 語を超えるレベル

#### d) 独立節と文断片の比率と平均ワード数

大きなテキストを分割して小さな単位として分析する場合、従来から文 (センテンス) という単位が用いられてきた。文はピリオドなどのパンクチュエーションによる区切りである。しかし文という区切りは特に会話で問題があるとされる (Biber & Conrad, 2009; Biber et al., 1999)。そのため本論文では文ではなく、文法的な区切りに基づく「節」 (clause) を用いる。これは動詞句を中心に作られる構造 (Biber et al., 1999) であり、このうち最も大きな単位が「独立節」 (independent clause) である。例えば「I eat an apple.」、下位の節が埋め込まれている「I eat an apple that I bought yesterday.」、節どうしが等位の「I eat an apple, but I do not drink coffee.」は3つともすべて独立節である。独立節は一般的に文とほぼ変わらない (Biber et al., 1999)。その節に対し、「文断片」 (sentence fragment) がある (注1)。これは動詞句から構成されない語の集まりである。新聞のヘッドライン、本のタイトル、図表、会話などに見られる。つまり動詞句を含むかどうかで独立節あるいは文断片に分類できる。

本研究では独立節と文断片の比率をレジスター間で比べると同時に、平均ワード数も調べる。これにより独立節 (文) をつないで書かれたテキストなのか、文断片を頻繁に使ったテキストなのかが分かる。筆者らの経験上、UI 英語には文断片が多いと考えられる。というのも、メニューやボタンでは限られた狭いスペースに言葉を書くため、本のタイトルのように短くなるはずだからである。また、UI は書籍のように直線的に読むわけではなく、画面のさまざまな場所にあるメニューや見出しを見つけて操作する。そのため独立節ではなく、キーワードとなる単語を短く並べた文断片が多いはずである。つまり画面のさまざまな場所にある文字を素早く見つけて読むという状況が言語に現れるのではないだろうか。

#### e) 命令と疑問

独立節は機能的に、平叙 (Declarative)、疑問 (Interrogative)、命令 (Imperative)、感嘆 (Exclamative) の4つに分けられる (Biber et al., 1999)。会話レジスターにおいては「命令」が頻繁に使われるが、これは状況がインタラクティブだからである (Biber et al., 1999)。UI においても、ユーザーとソフトウェアが対話し (インタラクティブ)、命令が多く用いられるのではないかと想像できる。また対話という状況では命令と同様に「疑問」も発生する (Biber et al., 1999)。そのため、UI 英語では疑問も頻出するのではないかと予想できる。ここでは特に疑問と命令に注目し、言語的特徴として表出しているのか確認したい。

#### f) 省略

省略 (ellipsis) とは要素を省くことで、言語的 (linguistics) と状況的 (situational) の2種類の省略があり、前者は文中から復元できるが後者はテキスト外部にあるコンテキストから推測しないと復元できない (Biber et al., 1999)。例えば「He bought an apple, and <he> ate it.」という文 (山かつこ内は実際は省略) で、he は文中から復元できるため言語的な省略である。経験上、UI 英語には省略が多いという印象がある。そのため、他のレジスターと比較して実際に多いのか、多い場合はどのような種類の省略が多いのかを調べる。

#### g) プレースホルダー

ソフトウェア UI では静的なテキストを表示するだけでなく、何らかの処理結果を表示することが多い。つまり動的に生成され、プレースホルダーに入れられて表示される。例えば時刻、ユーザー名、エラー・メッセージである。基本的にソフトウェアにしかないので、レジスター間の比較は行わない。これを言語的特徴と言ってよいのか判断が難しいが、ローカリゼーション時に翻訳者が目にするものであるため、どの程度出現するのか調査したい。

まとめると、語レベルで (a) 法助動詞、(b) 人称代名詞、(c) 指示代名詞/指示限定詞、語を超えるレベルで (d) 独立節と文断片の割合と平均ワード数、(e) 命令と疑問、(f) 省略、(g) プレースホルダーという7つの言語的な特徴を取り上げて分析する。語レベルの場合はコーパス全体で出現数をカウントし、1万ワードあたり頻度に換算できる。しかし語を超えるレベル (つまり d~g) では1つ1つ人間による確認が必要である (例えば命令か疑問か、など) ため、時間などの制約上コーパス全体について調べることは難しい。そこで1,000個のサンプルを取得して分析する。次にそのサンプル取得の方法を説明する。

### 2.4 分析単位への分割とサンプル取得

語を超えるレベルの分析を行うために、UI 英語 (PC およびモバイル)、新聞、ブログの各コーパスを手動で小さな単位に区切った。これを仮に分析単位と呼ぶ。この分析単位は前述の「独立節」および「文断片」を使用する。各コーパスのテキストを頭から目視し、独立節か文断片で区切った。このように分割後、UI 英語 (PC とモバイル)、新聞、ブログという4分野からそれぞれ1,000個ずつの分析単位 (独立節および文断片) をランダムに取得した。

### 3. 結果と考察

「2. 方法」で説明した方法を用いたところ、以下のような結果が得られた。同時に考察も行う。また、例文の下線は著者らによるものである。表に四角形 (■や□) で概数を示しているが、あくまで視覚的補助が目的で、正確な値は表内の数字である。

#### 3.1 法助動詞

9つの法助動詞について1万ワードあたりの出現頻度、および合計をカウントした (表3)。

	can	may	must	shall	will	could	might	should	would	合計
PC	110.2 ■■■■	18.9	23.3	0.0	37.5 ■	14.3	2.7	2.1	4.6	213.4
モバイル	82.0 ■■■	22.0	9.1	0.0	68.3 ■■	34.6 ■	3.4	1.4	15.0	235.7
新聞	10.6	5.9	2.8	0.0	43.3 ■	11.8	2.8	4.3	19.7	101.1
ブログ	40.1 ■	13.0	3.2	0.7	37.7 ■	13.0	5.3	9.9	30.6 ■	153.5

表3. 法助動詞の1万ワードあたり出現頻度 (■1つは25)

合計を見た場合、UI 英語レジスター (PC およびモバイル) は、新聞およびブログのレジスターと比べて法助動詞の出現数が多い。その中でも目立つのは「can」の多さで、PC およびモバイルの両方で第1位である。特にPC カテゴリーでは出現する法助動詞の約半分が can である。新聞やブログと比較した場合、PC と新聞では約10倍、モバイルとブログでも約2倍の差がある。そこでここでは can に注目したい。具体的に次のような例がある。

You can add or remove an account. (Outlook 2007)

Can not share this image file type. (Firefox OS)

The photo file cannot be downloaded. (iOS)

UI 英語では、ユーザーによる何らかの操作が「可能」あるいは「不可能」であることを示しているケースが多いように思える。ソフトウェアを利用する状況で、ユーザーは何らかの操作をして目標を達成したいはずである。肯定形である「can」の直後に出現する動詞 (be は除く) の頻度を調べてみると、use、take、change、add、view が多い。否定形である「can not/cannot」の直後では open、create、find、save、use が多い。これらはユーザーの操作を表す動詞である。そのうち否定形は、操作ができないというエラー状態を表しているのではないか。ユーザーが何かの操作をするとき (した後)、ソフトウェアから「できない」という応答が戻ってくるケースである。実際、HCI 分野ではエラー・メッ

ページの研究がされてきたと関連研究で述べた。「できる」や「できない」(エラー・メッセージ)は操作するという状況において重要な情報であり、それが **can** の頻度に現れているのではないかと。

### 3.2 人称代名詞

人称代名詞について1万ワードあたりの出現数をカウントした(表4)。方法で述べた通り、一人称は **I, we, me, us, my, our** の合計、二人称は **you, your** の合計、三人称は **he, she, they, him, her** (目的格)、**it, them, his, her** (所有格)、**its, their** の合計である。

	一人称	二人称	三人称
PC	13.0	220.9 ■■	43.0
モバイル	45.2	306.7 ■■■	50.4
新聞	72.4	26.4	334.4 ■■■■
ブログ	259.1 ■■	145.4 ■	334.8 ■■■■

表4. 人称代名詞の1万ワードあたり出現頻度 (■1つは100)

まず一人称から見てみる。表から分かる通り、他のレジスターと比較してブログで多く出現している。特にUI英語(PCとモバイル)との差が大きい。これはブログが日記や意見表明に使われている(Crystal, 2006)点に関係していると思われる。

続いて二人称である。UI英語で多く出現し、頻度は新聞の8倍以上である。ブログよりもやや多いが、新聞ほどとの差はない。次のような例がある。

Do you want to replace it? (Excel 2007)  
 Are you sure you want to continue? (iOS)  
 Please contact your system administrator. (Word 2007)  
Your USB storage is full. (Android)

最後に三人称である。これは新聞とブログで多く見られ、UI英語では少ない。新聞の事実伝達であっても、ブログの日記や意見表明であっても、頻繁に用いられる。

他のレジスターとの比較により、UI英語の特徴もはっきり分かる。UI英語では二人称が多い。これはユーザーがソフトウェアに処理命令を出し、ソフトウェアは結果を提示するというインタラクティブな状況であることが要因だと考えられる。特にソフトウェアがユーザーに呼びかける場合に **you**、ユーザーの所有物を指示する場合に **your** を使う。ユーザーを「the user」などと表現するより、**you** や **your** で直接表しているため、頻出するのだろう。

一方、一人称と三人称が相対的に少ない。一人称が少ない点を見ると、ユーザーが「I want to open this file.」などと自分をIで示して希望を伝えるというより、「Open this file.」といった命令で一方向的に直

接命令しているように思える。そういった点では人間同士のインタラクション（言葉遣いに配慮がある）とはやはり異なる。また三人称が少ないのは、新聞やブログのように代名詞を使いながら文を重ねる叙述が少ないからだと考えられる。後述するが、UI 英語における「文断片」の多さもこれと関係しているのではないだろうか。

### 3.3 指示代名詞／指示限定詞

1 万ワードあたりの指示代名詞および指示限定詞の出現数を比較する（表 5）。

	指示代名詞				指示限定詞			
	this	these	that	those	this	these	That	those
PC	8.9	0.2	0.3	0.2	123.0 ■■■■■	8.5	3.7	0.9
モバイル	12.9	0.3	0.6	0.1	124.6 ■■■■■	5.2	2.0	0.2
新聞	6.7	0.8	10.2	2.8	22.0	2.8	12.2	2.0
ブログ	28.2 ■	1.4	27.5 ■	7.0	29.9 ■	10.9	10.6	7.7

表 5. 指示代名詞および指示限定詞の 1 万ワードあたり出現頻度（■1 つは 25）

まず指示代名詞である。UI 英語（PC とモバイル）を見ると、**this** はブログと新聞との中間程度の頻度だが、**that** と **those** は少ない。例えば UI 英語では「Don't show **this** again」（Windows Phone 7.5）といったサンプルがある。

次に指示限定詞である。注目すべきは、UI 英語（PC とモバイル）において **this** が非常に多い点だろう。1 万ワードあたり 100 回以上出現し、新聞やブログの 4~5 倍程度の頻度で出現している。以下にいくつか例を挙げる。画面に表示される要素（dialog box、file、password）、ユーザーが用いるハードウェア（phone）、ユーザー操作（operation）などが見られる。

Close this dialog box when download completes (Media Player 10)

Cannot find this file. (Outlook 2007)

There is not enough memory to complete this operation. (Word 2007)

You can use this phone for emergency calls only. (Windows Phone 7.5)

This password is too long. (iOS)

指示代名詞および指示限定詞の全体を見た場合、UI 英語では「指示限定詞の **this**」が多い点が最も大きな特徴である。これはユーザーとソフトウェアが状況的なコンテキストを共有しているため、名詞を「**this**」で指示できるからであると考えられる。例えば、画面上でクリックしたファイルを「**this file**」と言ったり、ユーザーが手に持っている電話機を「**this phone**」と表現したりすることが可能となる。





## a) 命令

UI 英語 (PC とモバイル) では、1,000 サンプル中、共に 20%ほどが命令の独立節であった。新聞やブログと比べてかなり高い割合である。新聞の 100 倍、ブログの 5 倍ほど出現している。ユーザーとソフトウェアとの間に対話 (インタラクティブな状況) があることと関係していると考えられる。例えば LGSWE では、新聞レジスター (および学術文レジスター) はインタラクティブではないため命令節も疑問節も少ないとしている (Biber et al., 1999)。実際、表 9 で新聞の数字を見ると命令も疑問も非常に少ない。UI 英語における命令の多さは、新聞にはないインタラクティブ性が 1 つの要因であろう。

UI 英語の具体例を見る。UI 英語における命令は、誰に対する命令かという視点から大きく 2 つに分けられる。すなわち、ユーザーからソフトウェアに対する命令と、ソフトウェアからユーザーに対する命令である。前者のソフトウェアに対する命令の数をカウントしたところ、PC で 141 件 (141/195 = 72.3%)、モバイルで 148 件 (148/206 = 71.8%) であった。つまり多くが「ユーザーからソフトウェアに対する命令」である。以下に例を紹介する。

<ソフトウェアに対する命令>

Don't display this list on other computers I use (Outlook 2007)

Cancel (Android)

Find item that matches text (iOS)

<ユーザーに対する命令>

Follow the instructions in the wizard. (Excel 2007)

To save your changes, click the Microsoft Office Button and then click Save As. (Outlook 2007)

Please try again later. (iOS)

ソフトウェアに対する命令とユーザーに対する命令で、平均ワード数を調べた (表 10)。

	ソフトウェアに対する命令	ユーザーに対する命令
PC	3.55 ■■■	10.1 ■■■■■■■■■■
モバイル	2.51 ■■	7.38 ■■■■■■

表 10. UI 英語における、命令相手別の平均ワード数 (■1 つは 1 ワード)

ソフトウェアに対する命令の方が短く、ユーザーに対する命令の方が長い傾向がある。ソフトウェアに対する命令はメニューやボタンといった形で狭いスペースに記述される一方、ユーザーに対する命令はエラー・メッセージやヘルプなど比較的スペースに余裕がある場所に記述される点がひとつの理由だろう。また後者では、相手が人間であるため詳しく説明しているのではないか。いずれの場合でも PC よりモバイルの方が短い。モバイルでは画面が小さいため、短いメッセージで表示しようとしているのかもしれない。

何かを命令する場合、「please」を付けて丁寧さを表すことがある。人間がソフトウェアに丁寧に接する必要はないため、ソフトウェアに対する命令に please はない。一方、ソフトウェアからユーザーに対する命令の場合、please が付くケースがある。PC の場合 54 件中 1 件 (1/58 = 1.7%)、モバイルの場合 58 件中 4 件 (4/58 = 6.9%) あった。人間とソフトウェアとの対話 (インタラクション) があると言っても、丁寧さという部分では非対称である。

このように同じ命令と言っても、ソフトウェアに対する命令とユーザーに対する命令とでは、言語的な表現が異なる。前者の場合は短く、かつ丁寧さは求められないが、後者の場合は長め (詳細で) で丁寧な表現が加えられることがある。

## b) 疑問

疑問は PC とモバイル (UI 英語) でほぼ同程度であった (表 9)。UI 英語よりもブログの方が多い。LGSWE では、疑問節は対話の状況で発生しやすい (Biber et al., 1999) としているが、ブログほどには出現していないのである。同様に、対話という状況ではあるにも関わらず、命令よりも出現頻度は大幅に少ない。UI 英語での具体例をいくつか見てみる。

Do you want to configure it now? (Outlook 2007)

Are you sure you want to enable Wi-Fi? (iOS)

Would you like to close your existing tabs before turning off Private Browsing? (iOS)

今回取得した合計 46 件 (PC が 24、モバイルが 22) のうち、すべてが「ソフトウェアからユーザーに対する質問」であった。この疑問節は構文のおよび内容的にいくつかのパターンに分けられる (表 11)。

構文パターン	件数 (割合)	
Do you want to <動詞> ?	17 (40.0%)	ユーザーの希望の確認
<動詞> ?	11 (24.0%)	「Do you」などが省略された疑問節
Are you sure you want to <動詞> ?	8 (17.4%)	操作の再確認
Would you like to <動詞> ?	5 (10.9%)	ユーザーの希望の確認
その他	4 (8.7%)	

表 11. UI 英語における疑問の構文パターン

つまり、ソフトウェアからユーザーに対して希望の確認をしたり、ユーザーの操作を再確認したりといった対話で疑問が使われていることが分かる。前述の命令の多さと併せて考えると、ユーザーの「命令」に対してソフトウェアが「疑問」で確認するという対話の流れが想像できる。また「Do you」などの省略が見られるが、省略については次で論じる。

### 3.6 省略

1,000 サンプルのうち、少なくとも1つの省略があった件数は表12のとおりである。省略はどのレジスターでも発生し、件数自体は大きくは変わらない。

UI 英語		比較対象のレジスター	
PC	86	新聞	67
モバイル	98	ブログ	92

表12. 少なくとも1つの省略を含むサンプルの件数

これらの省略を言語的および状況的に分類した。表13に割合を示す。UIでは状況な省略がほとんどであるが、逆に新聞では言語的な省略がほとんどである。ブログはその中間である。つまり、UI英語は「状況的な省略」が多い点が大きな特徴である。

	言語的	状況的	両方 (1 サンプルに両方)
PC	12.8% ■	87.2% □□□□□□□□	0.0%
モバイル	10.2% ■	88.8% □□□□□□□□	1.0%
新聞	97.0% ■■■■■■■■■■	3.0%	0.0%
ブログ	43.5% ■■■■	55.4% □□□□□	1.1%

表13. 省略の種類（言語的または状況的）の割合（■または□1つは10%、四捨五入）

UI英語の状況的な省略について詳しく分類する。1,000 サンプル中、PCで75件、モバイルで87件、計162件の状況的な省略があり、以下のa~dに分類した。それぞれの具体例と、省略されていると思われる言葉を「→」の後に記した。また割合も計算した（図1）。

#### a) 主語の省略

PCで43件、モバイルで18件、合計61件ある。状況的な省略のうち「37.7%」を占める。

Cannot add a follow-up flag to this item. (Outlook 2007) → You など  
 Removes all similar formatting (Word 2007) → Clicking this button など  
 Can't access SD card. (Android) → The system や The application など

#### b) 動詞（助動詞も含む）の省略

PCで3件、モバイルで25件、合計28件ある。状況的な省略のうち「17.3%」を占める。

Printer Available (iOS) → is など

No apps found. (Windows Phone 7.5) → are や have been など

c) 主語と動詞の省略

PC で 27 件、モバイルで 40 件、合計 67 件ある。状況的な省略のうち「41.4%」を占める。

Use it anyway? (Excel 2007) → Do you など

Not a valid embedded object. (Word 2007) → This is など

Done (iOS) → The operation has been など

d) 目的語の省略

PC で 2 件、モバイルで 4 件、合計 6 件ある。状況的な省略のうち「3.7%」を占める。

Click to find out more. (Media Player) → here や this button など

Delete From My iPhone (iOS) → this や this item など

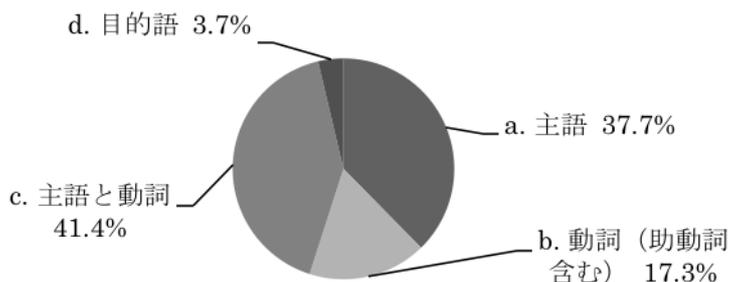


図 1. UI 英語 (PC およびモバイル) における「状況的な省略」の割合

UI 英語で発生する状況的な省略は「主語」、「動詞」、「主語と動詞」でほぼすべてを占める。状況的な省略が多いのは、ユーザーがテキストだけではなく、画面 (つまりコンテキスト) を見ながら操作しているためだと思われる。例えばメニューを開いてそこに「Removes all similar formatting」という説明があるなら、そのメニュー操作などが主語 (例えば *Selecting this menu*) であると想像できる。目的語の省略も同様で、操作する際にボタン上やリンク上に「Click to find out more.」と表示されれば、目的語はクリックする対象であるボタンやリンクであるとすぐに分かる。このようにユーザーはコンテキストを把握しているため、省略をしても問題ない場合は省略が発生するのだと考えられる。コンテキスト共有という状況が「省略」という言語的特徴に現れているのである。

ただし、省略は可能だとしても、必ずしもする必要はない。なぜ省略が発生するのだろうか。いくつか理由はあられると思われるが、1 つは「表示スペースの節約」だと思われる。例えばボタンやメニューには長いセンテンスは書けないため、省略する動機となるはずだ。また画面が小さいモバイルの場合、スペース節約の動機はさらに強くなる。つまり省略は、コンテキストを共有しているという状況下で、表示スペースを節約したいという意図から発生しているのではないかと想像する。こういった

部分はローカリゼーション時にも配慮し、翻訳者はレジスターに合った適切な訳文を作成する必要があるだろう。

### 3.7 プレースホルダー

本研究のコーパスでは、プレースホルダーをすべて「variablehere」に置換して扱っている。元々のプレースホルダーは以下のサンプルのように書かれている。「→」の後にプレースホルダーに入ると思われる情報を記しておく。

#d# new messages (Android) → 新規メッセージの数が「#d#」に  
 Do you want to connect to "#s#"? (Android) → ネットワークの接続先が「#s#」に  
 Sending %1\$@ of %2\$@ (iOS)  
 → 送信する全メッセージ数が「%2\$@」、送信中のメッセージ数が「%1\$@」に  
 You must enter an integer between 1 and {0}. (Excel 2007) → 整数が「{0}」に

1,000 サンプル中、少なくとも1つのプレースホルダーを含むものは、PC カテゴリーで97個、モバイルのカテゴリーで114個あった。つまり約10分の1程度のサンプルに1つ以上のプレースホルダーが含まれている。比較的頻繁に出現すると言えるだろう。そのため、翻訳者はプレースホルダーに入る文字列を想像できる能力を身に付ける必要がある。

## 4. 結論

ローカリゼーションでは「英語化」のニーズが存在する。そのため目標言語としての英語の特徴が明らかになれば、ローカリゼーション教育で有用である。本研究ではソフトウェア UI の英語についてレジスター分析を実施し、言語的特徴の一部を明らかにした。

新聞およびブログのレジスターと比較した場合、UI 英語で頻出する語や表現がある。語レベルでは、まず法助動詞で「can」の出現頻度が多い。これはユーザー操作が可能または不可能（エラー・メッセージ）であることを知らせる表現が多いからだと考えられる。次に、人称代名詞では二人称が多い。これは対話時にユーザーを「you」、ユーザーの所有物を「your」と呼んでいるからだと推測できる。また、指示限定詞の「this」が多い。これはユーザーとソフトウェアがコンテキストを共有しているため、名詞を this で指示できるからである。二人称および this が頻出することを考えると、共有されたコンテキストの中でユーザーに直接語りかけるというコミュニケーションが想像できる。

次に語を超えるレベルでは、まず「文断片が多く、かつ短い」と言える。これは画面上のさまざまな場所にあるメニュー名やボタン名などをすぐに見つけられるよう、短い文断片で表現しているからではないか。次に「命令」が多い点が挙げられる。また省略が頻出し、大部分が「状況的な省略」である。これはユーザーとソフトウェアがコンテキストを共有している状況で生じる言語的特徴であり、表示スペースの節約という動機から省略しているとも考えられる。指示限定詞 this が多いと述べたが、UI 英語を翻訳したり書いたりする場合は、コンテキスト共有を前提として省略したり、this で指示し

たりすることが可能である。最後に、「プレースホルダー」の出現も UI 英語の大きな特徴である。翻訳者はプレースホルダーにどのような文字が入るのか想像できる能力を身に付ける必要がある。

従来の翻訳研究における「レジスター分析」は基本的に、レジスター不明のテキストを分析して翻訳作業に役立てるといった目的で用いられてきた。本研究では、何らかのレジスターのテキストを定量的に比較し、その言語的特徴を記述する方法を用いた。これは翻訳研究における新しい手法として利用できるのではないだろうか。このような方法で既存のレジスターを分析しておけば、翻訳やローカリゼーションの教育で利用できると考える。

.....

### 【著者紹介】

西野 竜太郎 (NISHINO Ryutaro) フリーランス翻訳者、ソフトウェア開発者。東京工業大学大学院博士後期課程に在籍。著書に『アプリケーションをつくる英語』(インプレスジャパン/達人出版会、2012年)。連絡先: ryutaro@nishinos.com

野原 佳代子 (NOHARA Kayoko) 東京工業大学留学生センター/社会理工学研究科教授、専門は言語学、翻訳理論、サイエンスコミュニケーション。著書に『ディスカッションから学ぶ翻訳学』(三省堂、2014年)、『大学生のための社会人入門トレーニング コミュニケーション編』(共著、三省堂、2011年)、『科学技術コミュニケーション入門』(共著、培風館、2009年)など。

.....

### 【注】

1) LGSWE では non-clausal material という用語を使っている。文法的に区切るならこちらの方が適切だが、用語としてより認知されていると思われる「文断片」を使用する。

### 【参考文献】

- American National Corpus Project. (n.d.). MASC. Retrieved January 20, 2013, from <http://www.anc.org/data/masc/>
- android Git repositories. (n.d.). Retrieved July 31, 2012, from <https://android.googlesource.com/>
- Anthony, L. (2014). Antconc (Version 3.4.1) [Computer Software]. Tokyo, Japan: Waseda University. Retrieved from <http://www.antlab.sci.waseda.ac.jp/>
- Apple (n.d.). Apple Developer. Retrieved September 30, 2013, from <https://developer.apple.com/>
- Baker, M. (2011). *In Other Words: A Coursebook on Translation* (2nd ed.). Routledge.
- Biber, D., & Conrad, S. (2009). *Register, Genre, and Style* (eBook). Cambridge University Press.
- Biber, D., Johansson, S., Leech, G., Conrad, S., & Finegan, E. (1999). *Longman grammar of spoken and written English*. Longman.
- Crystal, D. (2006). *Language and the Internet* (2nd edn., eBook). Cambridge University Press.
- Esselink, B. (2000). *A Practical Guide to Localization*. John Benjamins Publishing.
- Facebook. (n.d.). Retrieved July 30, 2014, from <https://www.facebook.com/>
- Globalization and Localization Association. (n.d.). Localization Definitions. Retrieved July 01, 2014, from <http://www.gala-global.org/view/terminology>

- Hatim, B., & Mason, I. (1990). *Discourse and the Translator*. Longman.
- House, J. (2006). Text and context in translation. *Journal of Pragmatics*, 38(3), 338–358.
- Jiménez-Crespo, M. (2009). Conventions in localisation: a corpus study of original vs. translated web texts. *The Journal of Specialized Translation*, 12, 79–102.
- Jiménez-Crespo, M. (2013). *Translation and Web Localization* (eBook). Routledge.
- Leech, G. (2007). New Resources, or Just Better Old Ones? The Holy Grail of Representativeness. *Corpus Linguistics and the Web*, 133–149.
- McEnery, T., & Hardie, A. (2012). *Corpus Linguistics: Method, Theory and Practice* (eBook). Cambridge University Press.
- Microsoft. (n.d.). Microsoft Language Portal. Retrieved August 01, 2012, from <http://www.microsoft.com/Language/en-US/Translations.aspx>
- Microsoft. (2013). Microsoft by the Numbers. Retrieved November 06, 2013, from [http://www.microsoft.com/en-us/news/bythenumbers/ms\\_numbers.pdf](http://www.microsoft.com/en-us/news/bythenumbers/ms_numbers.pdf)
- Mozilla. (n.d.). Mozilla Cross-Reference. Retrieved December 27, 2013, from <http://mxr.mozilla.org/>
- Nielsen, J. (1994). *Usability Engineering*. Academic Press.
- Nishino, R., & Nohara, K. (2013). Characteristics of UI English: From Non-native's Viewpoint. In P. L. P. Rau (Ed.), *Cross-Cultural Design. Methods, Practice, and Case Studies* (pp. 323–332). Springer Berlin Heidelberg.
- O'Hagan, M., & Mangiron, C. (2013). *Game Localization* (eBook). John Benjamins.
- Pym, A. (2003). Redefining Translation Competence in an Electronic Age. In Defence of a Minimalist Approach. *Meta: Journal Des Traducteurs*, 48(4), 481–497.
- Shneiderman, B. (1979). Human Factors Experiments in Designing Interactive Systems. *Computer*, 12(12), 9–19.
- Sinclair, J. (2004). Corpus and Text - Basic Principles. In M. Wynne (Ed.), *Developing Linguistic Corpora: a Guide to Good Practice* (pp. 1–16). Oxford: Oxbow Books. Retrieved from <http://www.ahds.ac.uk/guides/linguistic-corpora/chapter1.htm>
- ハリデー, M. A. K., & ハッサン, R. (1991). 『機能文法のすすめ』(筧壽雄・訳). 大修館書店. [原著: Halliday, M.A.K., & Hassan, R. (1985). *Language, context, and text: Aspects of language in a social-semiotic perspective*. Deakin University Press.]
- 山田優 (2007). 「ローカリゼーションにおける翻訳と翻指理論研究」『翻指研究への招待』第1号: 57–68.
- 山田優 (2013). 「ローカリゼーション」鳥飼玖美子・編『よくわかる翻指通訳学』pp.68-69 ミネルヴァ書房.

