

# 多言語自動翻訳掲示板の利活用に関する実践研究

早稲田大学人間科学部人間情報科学科  
西村研究室

学籍番号：1J20F037  
名前：奥村飛悠

2024 年 1 月 7 日

# 目次

第 1 章	はじめに	1
1.1	現状 . . . . .	1
1.2	翻訳技術の進歩 . . . . .	2
1.3	既存サービスと先行研究 . . . . .	2
1.4	本研究の概要と目的 . . . . .	3
第 2 章	多言語自動翻訳掲示板について	4
2.1	システムの概要 . . . . .	4
2.2	主な機能と利用フロー . . . . .	4
2.3	システム構成の概要 . . . . .	13
2.4	コンポーネント設計 . . . . .	20
2.5	コンポーネントの詳細 . . . . .	21
2.6	データ設計 . . . . .	25
第 3 章	利活用についての分析	28
3.1	ユーザー獲得 . . . . .	28
3.2	利活用の結果 . . . . .	28
3.3	システム利用の実際と課題 . . . . .	30
	参考引用文献	32

# 第 1 章

## はじめに

### 1.1 現状

インターネットの普及やソーシャルメディアの台頭により，オンラインでのコミュニケーションが一般的になっている（大向，2006）．その中でも，誰でも気軽に参加することのできる掲示板は重要なコミュニケーションの場となっている．日本では，”5ちゃんねる”（サイト名を”2ちゃんねる”から”5ちゃんねる”へ 2017 年 10 月に変更）が広く知られている一方で，米国発の”reddit”は国際的に認知度が高く，日別のアクティブユーザー数は 5700 万人，総投稿数は 130 億投稿を超えている（Reddit Inc, 2023）．これらの大規模な掲示板は情報の集積場所として，またユーザー間の活発な議論の場として重要な役割を果たしている．

しかしながら，掲示板は誰もが利用できるコミュニケーションの場であるにも関わらず，現状ではそのコミュニケーションは主に同一言語間で行われている．具体的には，”5ちゃんねる”では主に日本語，”reddit”では主に英語が使用されている．そのため，異なる言語を使用するユーザーは，翻訳ツールや外部の翻訳サービスを頼るか，専用のスレッドや言語コミュニティを探すことが一般的となっている．しかし，これらの方法にはデメリットが存在する．例えば，翻訳ツールを利用すると時間と手間がかかるため，掲示板の持つ即時性や気軽さというメリットを十分に享受することが難しくなる．

同一言語間でのコミュニケーションが主流となっている背後には複数の要因が考えられるが，その一つとして翻訳技術の品質が十分でなかったことが考えられる．2004 年には「コミュニケーションツールとして使用する場合に十分な品質を持っているとはいえない」（船越ほか，2004）との指摘があり，さらに 2010 年にも「近年，翻訳技術は急速に進展しているが，高精度な翻訳を行うことは困難である．コミュニケーションにおいて，不適切な翻訳箇所を含む文章は話者間の相互理解を困難にし，円滑なコミュ

ニケーションの妨げとなる」(宮部ほか, 2010)とも指摘されている。また, 多言語間でのコミュニケーションにおいては, 翻訳の品質が極めて大きな影響力を持つことも確認されている(船越ほか, 2004)。このように, コミュニケーションに大きな影響を与える翻訳技術の品質が十分でなかったため, ユーザーは翻訳技術を利用して会話の中心となっている言語以外を使用してまで会話を試みなかったのではないだろうか。

## 1.2 翻訳技術の進歩

しかしながら, 翻訳サービスの精度は日々向上している。これについて, 「近年, Google 翻訳や DeepL, そしてページ全体翻訳機能の進化が著しい」(村本, 2022)との報告があり, その背景には機械学習の進歩が影響を与えている。「Google 英日翻訳が NMT (ニューラル機械翻訳)を採用したことで, 目標言語の流暢さが格段に向上した」(影浦, 2017)との報告がある。同様に NMT を採用している DeepL は, 2017 年にサービスを開始し, その高品質な翻訳サービスが評価されている(亀田, 2022)。さらに「2020 年と 2021 年には, 文章の意味をより正確に伝えられ, 業界特有の専門用語もうまく処理できる新たなモデルを発表」(DeepL, 2023)している。これらのことから, 翻訳サービスの精度は日々向上されていることが分かる。

また, 多くの翻訳サービスが開発者向けに API を提供している。その代表例としては, Google Cloud の Translation AI (Google Cloud, 2023)や DeepL API (DeepL, 2023)がある。これらのサービスを開発者が利用するための便利なライブラリも存在している。具体的には, Google Translate API (現在の Translation AI)を利用するための Python のライブラリである googletrans (PyPI, 2023a)や deepl (PyPI, 2023b)がある。このような API やライブラリの存在により, 開発者は翻訳機能を自身のサービスに容易に組み込むことが可能となっている。

## 1.3 既存サービスと先行研究

過去には, "enjoy Korea"という日本語と韓国語の翻訳機能を持つ掲示板サービスが存在していたが, 利用率の低下を理由に 2009 年 6 月 8 日にサービスを終了している(野津, 2009)。また, 小川ら(2009)は日本語とウイグル語間の翻訳掲示板システムを開発している。しかし, 彼らの研究は主にシステムの開発に焦点を当てており, システムを使用するユーザーのデータ収集やその分析までには至っていない。

一方, 藤井ら(2005)はアノテーションや折り返し翻訳に着目し, 中国語, 韓国語, 日本語間の翻訳 BBS である "AnnoChat"を開発した。翻訳の精度がコミュニケーションの理解度に影響を与える可能性を示しているが, ユーザー同士の具体的なコミュニケー

ションの内容までは調査していない。また、吉野ら（2006）はユーザインタフェースのカスタマイズ性に焦点を当てた研究を行い、“CustomChat”というシステムを開発したが、これも具体的なチャットの内容などについては触れられていない。

これらの事例や研究を見ると、多言語間のコミュニケーションを可能にする翻訳掲示板に関する研究やサービスは確かに存在している。しかし、それらは主にシステムの開発や翻訳の精度と理解度の関係性、ユーザインタフェースの改良に焦点を当てており、異なる言語を使用するユーザーがシステムをどのように使うのか、どのようなコミュニケーションが起こるのかという点については、まだ十分に研究されていないと言える。

## 1.4 本研究の概要と目的

これらの背景から、本研究では、掲示板のグローバル化を進めるため、近年の高精度な翻訳サービスを利用した多言語自動翻訳掲示板の開発とその利活用について実践的な研究を行う。我々が提案する多言語自動翻訳掲示板では、ユーザーは表示言語を選択することにより、選択した言語で掲示板の投稿を閲覧することを可能にする機能をつける。これにより、異なる言語を使用するユーザー間でも、自由なコミュニケーションが促進され、掲示板の持つ即時性や気軽さというメリットを維持することができる。

そして、この掲示板をインターネット上に公開し、使用者から得られるデータを収集する。その後、得られたデータを分析し、多言語自動翻訳掲示板がユーザーのコミュニケーションにどのような影響を与えるのか、多言語自動翻訳掲示板上で異なる言語を使用するユーザー同士がどのようなコミュニケーションをするのかを評価する。具体的には、ユーザー間のコミュニケーション量や内容、トピックの多様性、言語間のコミュニケーション方法などを指標として用いる。

我々の研究は、新たな掲示板の形を示すだけでなく、機械翻訳技術とその実用化の進歩に貢献することを期待している。本研究の結果が、ユーザーが自由に多言語コミュニケーションを享受できるインターネットの環境整備に向けた一歩となることを願っている。

## 第 2 章

# 多言語自動翻訳掲示板について

### 2.1 システムの概要

本研究では、多言語自動翻訳掲示板である「The Channel」という Web アプリケーションを開発した。このシステムは、世界中のユーザーが自分の言語で投稿することができる。そして、その投稿は閲覧するユーザーが選択した言語に翻訳されて表示されるため、ユーザーは好きな言語でコンテンツを読むことができる。

システムの中核は、機械翻訳技術が担っている。これにより、ユーザーが投稿したテキストはリアルタイムで他の言語に翻訳され、多様なユーザーがアクセスできるようになる。例えば、日本語で書かれた投稿は、英語、スペイン語、中国語などに瞬時に翻訳され、異なる言語のユーザー間の交流を可能にする。

システムは、掲示板として必要最低限の機能を備えている。ユーザーは簡単にスレッドを作成することや、自分の母国語でコメントを投稿することができる。

セキュリティに考慮し、ユーザーの個人名やメールアドレス等を取り扱わないようにしている。

### 2.2 主な機能と利用フロー

掲示板は以下の機能を持つ。

- スレッド作成：新しいスレッドを作成する機能
- コメント投稿：既存のスレッドに対してコメントを投稿する機能
- 閲覧：スレッドの内容を表示する機能
- 言語選択：ユーザーが表示言語を選択する機能

図 1 にシステム利用フローを示す.

- (1) ユーザーはブラウザを通して掲示板を閲覧する.
- (2) ユーザーは閲覧しているスレッドに対してコメントを投稿することができる.
- (3) ユーザーは自由にスレッドを作成することができる.
- (4) ユーザーはいつでも言語を選択することができる.

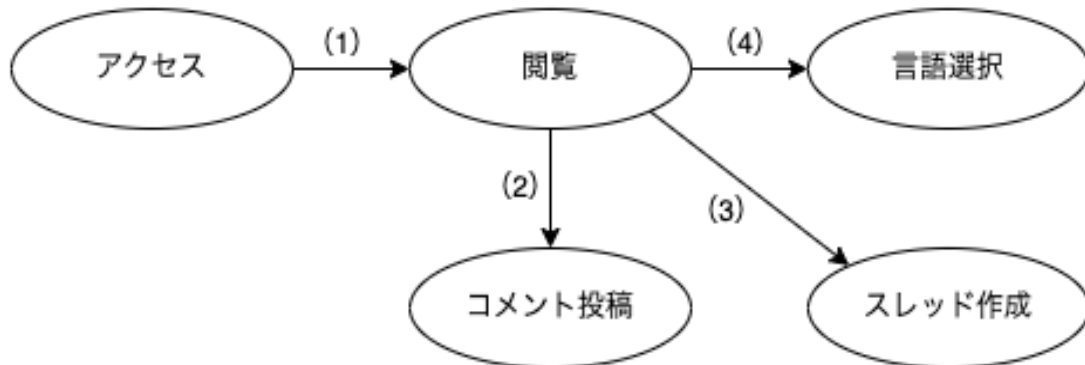


図 1: システム利用フロー

### 2.2.1 システムの画面

以下に、主要な画面の概要を示す.

#### トップページ画面

トップページ画面 (図 2) は、システムへの入口であり、ユーザーがスレッドを選択する主要な場である. この画面は、最新更新スレッドリストと新着スレッドリストが表示されている. 任意のスレッドを選択することで、スレッドの中身を閲覧することができる.



図 2: トップページ画面

## スレッド詳細画面

スレッド詳細画面 (図 3) では、選択されたスレッドの内容と、それに対するユーザーのコメントが表示される。ユーザーは、この画面の下部にある入力欄を使用して新たなコメントを投稿することが可能である。



≡ The Channel

言語  
Japanese(日本語) ▾

## 東京でやるべきことは何ですか？

こんにちは！私はアメリカの交換学生です。私はここに1年だけここにいますので、地元の人々がやりたいことを試してみたいです！東京や日本でお気に入りの場所は何ですか？それは何でもかもしれません！（多分観光客に知られていない場所）

1 NO NAME 2023-12-14 11:01:41UTC

日本によこそ！

ざうおという魚釣りできて、その場で釣った魚を食べられるお店があります。海外だとあまりないと思うので是非！

2 NO NAME 2023-12-14 11:02:13UTC

1 << <https://www.zaou.com/en/howto/>

3 ぶんぶん 2023-12-16 10:07:09UTC

日本に来てくれてありがとう。主に駅の中や、ホーム上に存在する『立ち食いそば屋』巡りなんてどうですか。 ＊『そば』アレルギーが無い人のみ推奨。<https://tachigui-soba.com/cooktag/ekisoba>

4 NO NAME 2023-12-19 05:20:12UTC

外国人は日本の焼肉に驚くという話を聞いたことある

5 NO NAME 2023-12-19 11:25:35UTC

山手線一周かなあ

6 NO NAME 2023-12-20 04:46:46UTC

水上バス、アニメイト、貸切温泉

作成者名

コメント

送信

プライバシーポリシー 利用規約

Copyright © 2023 早稲田大学 西村研究室

図 3: スレッド詳細画面

## スレッド作成画面

スレッド作成画面 (図 4) では、ユーザーが新たなスレッドを作成することができる。

≡ The Channel 言語 Japanese(日本語) ▼

### 新規スレッド作成

タイトル

作成者名

概要

スレッドを作成

[プライバシーポリシー](#) [利用規約](#)

Copyright © 2023 早稲田大学 西村研究室

図 4: スレッド作成画面

## 2.2.2 主要機能とその操作

以下では、各機能の詳細と画面操作を説明する。

### スレッド作成

スレッド作成画面 (図 4) では、新規のスレッドを作成することができる。スレッド作成欄 (図 5) にタイトルと作成者名と概要を入力し、「スレッドを作成」ボタンをクリックすることでスレッドが作成される。空欄の場合にはエラーが表示される (図 6)。

### 新規スレッド作成

タイトル

作成者名

概要

スレッドを作成

図 5: スレッド作成欄

## 新規スレッド作成

タイトル

タイトルは必須です

作成者名

作成者名は必須です

概要

概要は必須です

スレッドを作成

図 6: スレッド作成欄エラー

## コメント投稿

スレッド詳細画面 (図 3) では、既存のスレッドに対してユーザーはコメントを投稿することができる。コメント投稿欄 (図 7) に作成者名とコメント内容を入力し、「投稿」ボタンをクリックすることでコメントが投稿される。匿名投稿が可能である。コメントが空欄の場合にはエラーが表示される (図 8)。

作成者名

コメント

送信

図 7: コメント投稿欄

作成者名

コメント

|

コメントは必須です

送信

図 8: コメント投稿欄エラー

## スレッド詳細の閲覧

トップページ画面 (図 2) に表示されている最新更新スレッドリスト (図 9) または新着スレッドリスト (図 10) から任意のスレッドを選択し、中身を閲覧することができる。

### 最新更新スレッド

幽体離脱したことある人？  
更新日: 2024-01-02 11:43:09UTC

日本で一番美味しかったもの  
更新日: 2023-12-25 15:15:28UTC

国際カップル、国際結婚についてどう思う？  
更新日: 2023-12-25 14:52:56UTC

好きなアニメ何??  
更新日: 2023-12-23 19:31:17UTC

一人っ子vs兄弟 どちらが幸せか？  
更新日: 2023-12-23 19:29:04UTC

発熱があるときはそれをする必要があります  
更新日: 2023-12-23 19:21:40UTC

日本のナットについてどう思いますか？あなたの国では日本のナットのように扱われた同様の食べ物がありますか？  
更新日: 2023-12-23 14:29:28UTC

大谷翔平の移籍先はどこになるのか？  
更新日: 2023-12-20 08:29:18UTC

< 1 2 3 >

図 9: 最新更新スレッドリスト

## 新着スレッド

幽体離脱したことある人？  
作成日: 2023-12-25 15:22:21UTC

発熱があるときはそれをする必要があります  
作成日: 2023-12-23 19:21:40UTC

国際カップル、国際結婚についてどう思う？  
作成日: 2023-12-23 14:15:26UTC

好きなアニメ何??  
作成日: 2023-12-17 14:40:36UTC

日本で一番美味しかったもの  
作成日: 2023-12-17 07:16:51UTC

日本に留学したい人集まれ  
作成日: 2023-12-17 07:15:24UTC

『頑張る彼女を応援してもらえないか』 PRODUCE 101 JAPAN( <https://produce101.jp>)というオーディションに参加されているそう。日プというらしい。初めて聞いた。101名の中の唯一の早大生。ま、応援するよね(笑)  
【田中優希】さん  
作成日: 2023-12-17 05:29:58UTC

ドリンクバー  
作成日: 2023-12-16 12:13:22UTC

< 1 2 3 >

図 10: 新着スレッドリスト

## ヘッダ

ヘッダ (図 11) は、システム内のどの画面にいても常に上部に表示される。ヘッダには、サイトのロゴと言語選択メニュー、サイドメニューへのアクセスボタンが含まれている。



図 11: ヘッダ

## サイドメニュー

サイドメニュー (図 12) は、トップページとスレッド作成画面への遷移を提供する。ヘッダヘッダ (図 11) 左部のメニューボタンをクリックすることで表示される。

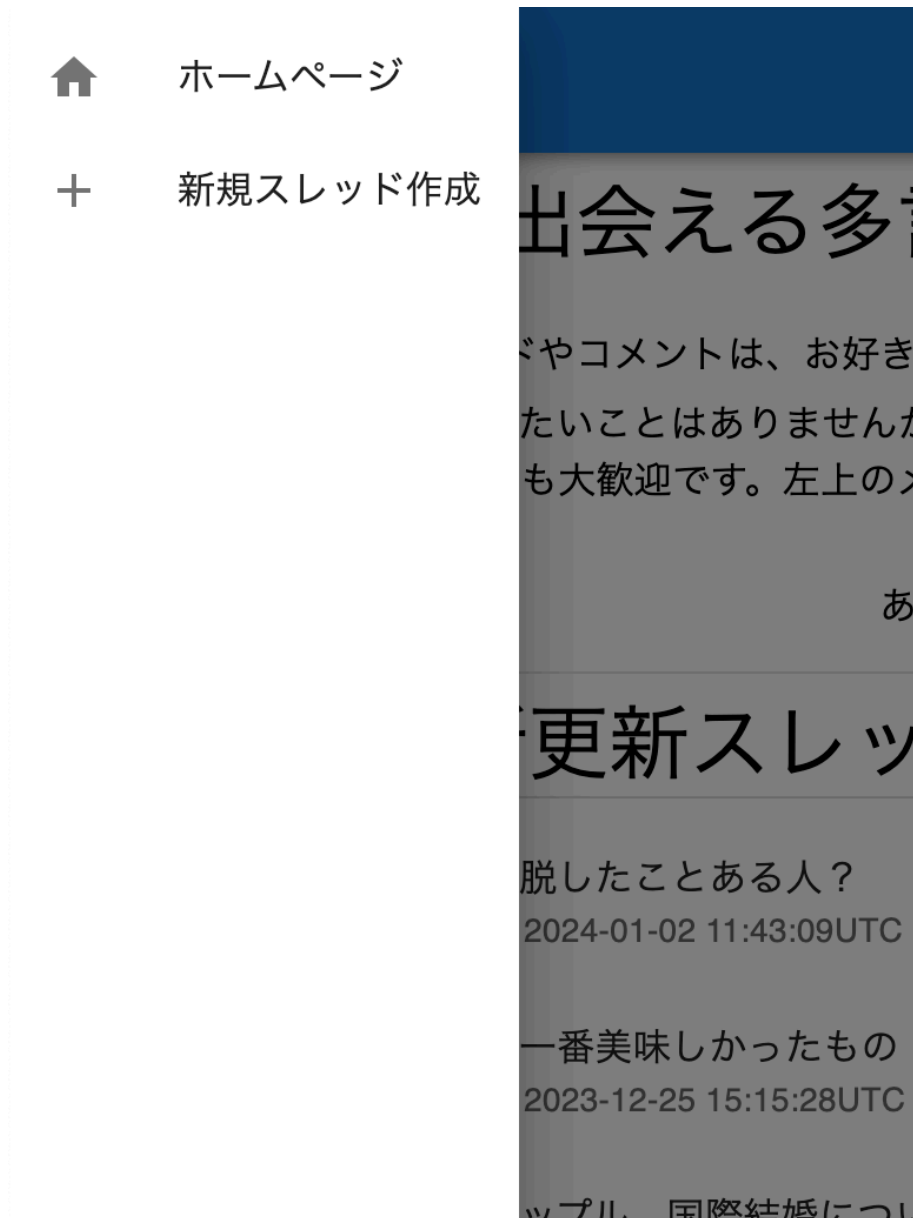


図 12: サイドメニュー

## 言語選択

ユーザーは言語選択欄 (図 13) から自由にシステムの表示言語を変更できる. ヘッダ (図 11) 右部にある言語選択欄の利用可能な言語の中から選択する. 選択された言語は, ユーザーが閲覧するコンテンツに適応される.

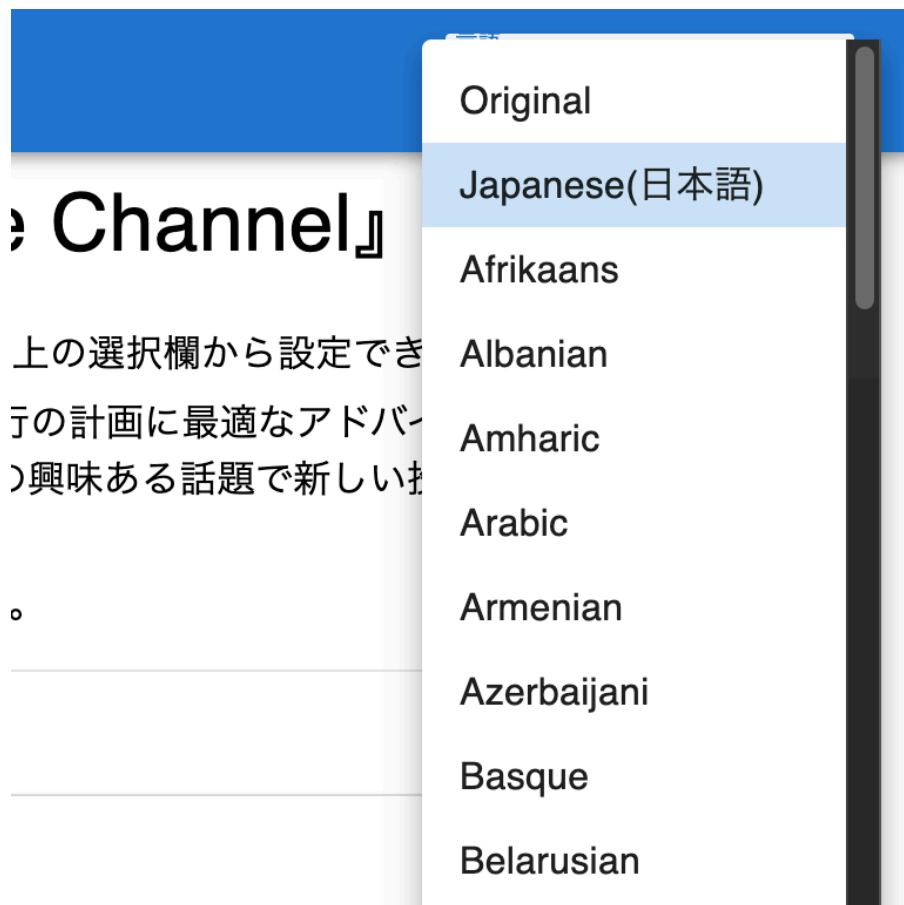


図 13: 言語選択欄

## フッタ

フッタ (図 14) は、システム内のどの画面にいても常に上部に表示される。フッタには利用規約やプライバシーポリシーへのリンクが含まれており、ユーザーはこれらの文書にアクセスすることができる。



図 14: フッタ

## 2.3 システム構成の概要

開発した多言語自動翻訳掲示板は、フロントエンド、バックエンド、データベース、および Web サーバーといったコンポーネントによって構成されている。図 15 にそれぞれ

のコンポーネントにおいて採用した主要な技術を示す技術スタック図を示す. フロントエンドの開発には言語として TypeScript, ライブラリとして React, フレームワークとして Next.js, UI コンポーネントフレームワークとして Material-UI, スタイルシートとして Sass を採用した. バックエンドには言語として Python, フレームワークとして FastAPI を採用した. データベース管理には MySQL を採用した. Web サーバーには Nginx を採用した. システムのコンテナ化には Docker を使用した. ソースコードの管理には Git, GitHub を使用した.

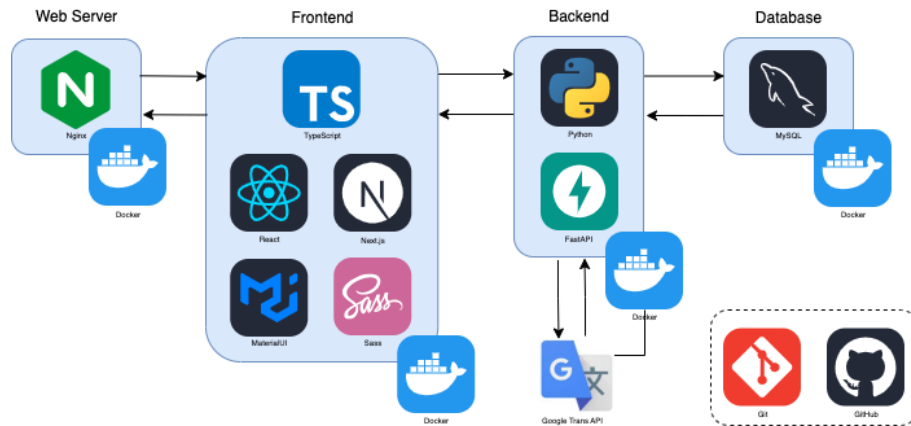


図 15: 技術スタック図

### 2.3.1 サーバー

本研究では, さくらの VPS をサーバーとして使用している. サーバーのオペレーティングシステムには Ubuntu 22.04.3 LTS を採用した. 以下にサーバーの CPU, メモリ, ストレージ, ネットワークインターフェイスに関する詳細な情報を記載する.

#### メモリ情報

サーバーのメモリは以下の通りである.

- 合計: 7937 MB
- 使用中: 1534 MB
- 空き: 860 MB
- バッファ/キャッシュ: 5541 MB
- 利用可能: 6094 MB
- スワップ: 0 MB



## ディスク使用状況

サーバーのディスク使用状況は以下の通りである.

- ファイルシステム:
  - **tmpfs**: 使用中 794 MB (使用中 1.4 MB, 空き 793 MB)
  - **/dev/vda2**: 使用中 788 GB (使用中 40 GB, 空き 708 GB)

## ディスク構成

サーバーのディスク構成は以下の通りである.

- ディスク名: vda
- サイズ: 800 GB
- タイプ: ディスク
- パーティション:
  - **vda1**: 1 MB
  - **vda2**: 800 GB (マウントポイント: /)

## CPU 情報

サーバーの CPU に関する情報は以下のとおりである.

- アーキテクチャ: x86\_64
- CPU 操作モード: 32-bit, 64-bit
- アドレスサイズ: 46 ビット物理, 48 ビット仮想
- バイト順序: Little Endian
- CPU 総数: 6
- オンライン CPU リスト: 0-5
- ベンダー ID: GenuineIntel
- モデル名: Intel Xeon Processor (Cascadelake)
- CPU ファミリー: 6
- モデル: 85
- コアあたりのスレッド数: 1
- ソケットあたりのコア数: 1
- ソケット数: 6

- ステッピング: 5
- **BogoMIPS**: 4199.99
- サポートされているフラグ: [フラグのリスト]
- ハイパーバイザのベンダー: KVM
- 仮想化タイプ: 完全仮想化
- キャッシュの合計:
  - L1d: 192KiB (6 インスタンス)
  - L1i: 192KiB (6 インスタンス)
  - L2: 24MiB (6 インスタンス)
- **NUMA ノード数**: 1
- **NUMA ノード 0 CPU**: 0-5
- 脆弱性と軽減策:
  - L1tf: PTE Inversion による軽減
  - Meltdown: PTI による軽減
  - Spectre v1: ユーザーコピー/swapgs バリア, ユーザーポインターのサニタイズによる軽減
  - Spectre v2: IBRS, IBPB 条件付き, STIBP 無効, RSB 充填, PBRSE-eIBRS Not affected による軽減

## ネットワークインターフェイスの状態

サーバーのネットワークインターフェイスの設定と状態は以下の通りである。

- **br-f4387cc8d030**:
  - IP アドレス: 172.18.0.1/16
  - MAC アドレス: 02:42:06:18:81:07
  - 受信パケット: 851,033 (2.0 GB)
  - 送信パケット: 893,012 (78.0 MB)
- **docker0**:
  - IP アドレス: 172.17.0.1/16
  - MAC アドレス: 02:42:bf:a4:bf:64
  - 受信パケット: 6,763 (625.9 KB)
  - 送信パケット: 7,761 (43.2 MB)

### 2.3.2 セキュリティ対策

本研究では、システムのセキュリティ強化のためにファイアウォール（UFW）と AIDE（Advanced Intrusion Detection Environment）を導入した。

#### ファイアウォール（UFW）の設定

本システムでは、外部からの不正アクセスを防ぎ、内部ネットワークのセキュリティを高めるためにファイアウォール（UFW）を設定している。ファイアウォールは、不正なネットワークトラフィックを検出しブロックすることで、システムを保護する重要なセキュリティ機能である。以下がその具体的な設定である。

- 状態: アクティブ
- ルール:
  - ポート 22/tcp: 全てのアドレスからのアクセスを許可
  - ポート 443/tcp: 全てのアドレスからのアクセスを許可
  - IPv6 に関しても同様の設定

このように設定することで、システムは不正なアクセスを効果的に阻止し、同時に必要な通信は許可することができる。結果として、セキュリティを損なうことなく、システムの柔軟性とアクセシビリティを維持している。

#### AIDE（Advanced Intrusion Detection Environment）の導入

本システムでは、ファイルシステムの整合性を監視するために AIDE を使用している。AIDE はファイル変更を検出するホストベースの侵入検出システムで、システムに不正な変更が生じた場合に警告を発する。

AIDE の設定は以下の通り。

- AIDE のスケジュール設定：システムは毎日午前 0 時に AIDE を実行し、ファイルシステムの整合性をチェックする。
- メール通知の設定：異常が検出された場合、システムは自動的に指定した E-mail アドレスに通知する。

このセットアップにより、システムの安全性を高め、不正アクセスや変更があった場合に迅速に対応できる。

### 2.3.3 バックエンド開発

本研究のバックエンド開発においては, Python 言語と FastAPI フレームワークを中心に, 多様なライブラリを利用した. Python はバックエンドの主要言語として, FastAPI は主要フレームワークとして採用した. また, 翻訳機能は googletrans ライブラリを用いて実現した.

以下に, 本研究において使用したライブラリの一覧を示す.

- anyio: 3.6.2
- cachetools: 5.3.1
- certifi: 2022.12.7
- chardet: 3.0.4
- click: 8.1.3
- fastapi: 0.95.1
- googletrans: 4.0.0rc1
- h11: 0.9.0
- h2: 3.2.0
- hpack: 3.0.0
- hstspreload: 2023.1.1
- httpcore: 0.9.1
- httptools: 0.5.0
- httpx: 0.13.3
- hyperframe: 5.2.0
- idna: 2.10
- mysql: 0.0.3
- mysql-connector-python: 8.0.33
- mysqlclient: 2.1.1
- protobuf: 3.20.3
- pydantic: 1.10.7
- python-dotenv: 1.0.0
- pytz: 2023.3
- PyYAML: 6.0
- rfc3986: 1.5.0
- sniffio: 1.3.0
- starlette: 0.26.1
- typing\_extensions: 4.5.0

- uvicorn: 0.22.0
- uvloop: 0.17.0
- watchfiles: 0.19.0
- websockets: 11.0.2

### 2.3.4 フロントエンド開発

本研究のフロントエンド開発においては，TypeScript 言語と React ライブラリ，Next.js フレームワークを中心に，Material UI を含む多様なライブラリを利用した。TypeScript はフロントエンドの主要言語として，React と Next.js は主要なライブラリとフレームワークとして採用した。また，UI コンポーネントは Material UI を用いて実現した。

#### 依存関係

フロントエンド開発において直接利用されたライブラリの一覧を示す。

- @emotion/react: 11.11.0
- @emotion/styled: 11.11.0
- @mui/icons-material: 5.11.16
- @mui/material: 5.13.2
- @types/gtag.js: 0.0.17
- @types/js-cookie: 3.0.3
- @types/moment: 2.13.0
- js-cookie: 3.0.5
- moment: 2.29.4
- moment-timezone: 0.5.43
- next: 13.3.1
- next-sitemap: 4.2.3
- normalize.css: 8.0.1
- react: 18.2.0
- react-dom: 18.2.0
- react-hook-form: 7.45.2
- sass: 1.62.1

## 開発用依存関係

開発プロセスを支援するために使用されたライブラリの一覧を示す.

- @types/node: 18.16.0
- @types/react: 18.0.38
- @types/react-dom: 18.0.11
- @typescript-eslint/eslint-plugin: 5.59.0
- @typescript-eslint/parser: 5.59.0
- eslint: 8.39.0
- eslint-config-next: 13.3.1
- eslint-config-prettier: 8.8.0
- eslint-plugin-prettier: 4.2.1
- prettier: 2.8.8
- typescript: 5.0.4

### 2.3.5 データベースについて

データベースとして MySQL が採用されている. このセクションでは, データベースの構成と, それがシステムにどのように統合されているかについて説明する.

### 2.3.6 通信の取り扱いと Nginx の役割

本システムにおける通信は https を通じて行われ, セキュリティを確保しつつユーザーのリクエストを効率的に処理する. ユーザーは初めに Nginx を経由してフロントエンドの画面にアクセスし, 画面表示に必要なデータを取得する. また, ユーザーからの POST アクセスは Nginx を通じてバックエンドに転送され, 処理される. このセクションでは, Nginx がシステム内で果たす重要な役割と, それが通信プロセスにどのように統合されているかについて詳述する.

## 2.4 コンポーネント設計

この掲示板システムは, 表示コンポーネント及びデータ更新コンポーネントという二つの主要な構成要素から成立している. 表示コンポーネントは, ユーザーインターフェースとして機能するブラウザとの通信を担当し, データ取得時にユーザーの言語情報を収

集する。これにより、ユーザーの言語に応じた適切なデータをブラウザに表示する。一方、データ更新コンポーネントは、主にユーザーによる投稿データの更新を管理する。このコンポーネントは、ブラウザから送信されるデータの検証と処理を行い、その後、データベースにデータを保存する。これにより、ユーザーのインタラクションに基づく動的なコンテンツの生成と管理が可能となる。

## 2.5 コンポーネントの詳細

### 表示コンポーネント

表示コンポーネントは、ユーザーインターフェイスの構成要素として機能し、以下のファイル構造によって定義されている。

ウェブページは、それぞれ独立したファイルによって構築される。

- `/pages/index.tsx` — トップページ
- `/pages/policy.tsx` — プライバシーポリシーページ
- `/pages/terms.tsx` — 利用規約ページ
- `/pages/404.tsx` — 404 エラーページ
- `/pages/thread/create.tsx` — スレッド作成ページ
- `/pages/thread/index.tsx` — スレッド詳細ページ

UI コンポーネントは再利用性を考慮して設計され、Atoms および Organisms の二つのカテゴリに分類される。Atoms は単純な UI 要素を、Organisms は複数の Atoms 要素を組み合わせて構成されるより複雑な UI セクションを形成する。

Atoms カテゴリには以下のコンポーネントが含まれる。

- `CustomTextField.tsx` — ユーザー入力を受け取るフィールド
- `MenuItem.tsx` — メニュー項目表示
- `TextWithNewLines.tsx` — 改行を含むテキスト表示

Organisms カテゴリは以下のコンポーネントを含む。

- `CookieBanner.tsx` — クッキーの使用に関するユーザーへの通知バナー表示
- `Footer.tsx` — フッター表示
- `Header.tsx` — ヘッダー表示
- `Meta.tsx` — ウェブページのメタデータ
- `Pager.tsx` — ページャー表示

- ThreadList.tsx — スレッドリスト表示

図 16 は、表示コンポーネントの動作プロセスを示したシーケンス図である。

(1) ブラウザはサーバーに対し、ページ固有データの要求を行う。ページ固有データには、ページタイトル、ボタンのラベル、利用規約、免責事項といった静的情報が含まれている。これらの情報は多言語に対応しており、ブラウザ及びユーザーの言語設定に基づいて提供される。翻訳 API を使用しないことにより、システムの処理時間の短縮が図られている。ページ固有データの取得後、ブラウザは画面を構築し、ユーザーに対して表示を行う。

(2) ブラウザは、ページ表示のための動的データをサーバーに要求する。動的データが取得されるまでの間、スケルトンスクリーンを用いて表示がなされる（図参照）。プロセス (3) および (4) が完了すると、ページデータがブラウザに返送される。

(3) サーバーは、ブラウザからのリクエストを解析し、データベースからデータを取得する。取得したデータはブラウザに返す形式に変換される。

(4) サーバーは、ブラウザの選択した言語情報に従って内容を翻訳する。そのために外部の翻訳 API にリクエストを送信し、返された翻訳結果をブラウザに返す。



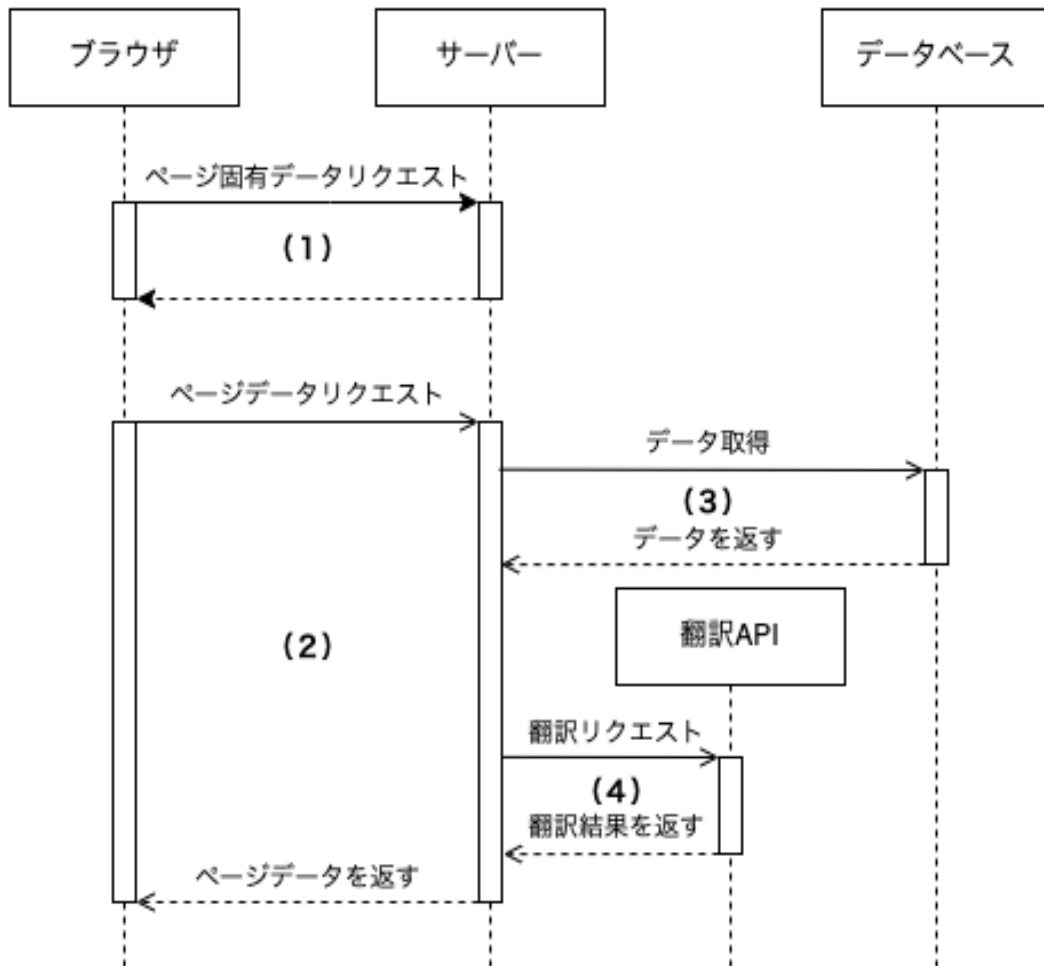


図 16: 表示コンポーネントのシーケンス図 (1) のとこ直す, ページを表示すること書く?

## データ更新コンポーネント

データ更新コンポーネントは以下のような構成になっている。

- `thread_router.py` … スレッド関連のルーティングを管理
- `thread_application.py` … スレッドのアプリケーションロジックを処理
- `thread_infrastructure.py` … スレッドのインフラストラクチャ関連処理
- `comment_router.py` … コメント関連のルーティングを管理
- `comment_application.py` … コメントのアプリケーションロジックを処理
- `comment_infrastructure.py` … コメントのインフラストラクチャ関連処理
- `translation.py` … 翻訳機能の実装

上記のファイルリストは, 以下のアーキテクチャを採用し作成された. このアーキテク

チャは、明確な役割と責任を持つ複数のレイヤーに分かれている。

### ■ルーターレイヤー

`thread_router.py` と `comment_router.py` はルーターレイヤーに属し、外部からのリクエストを適切な処理ロジックに振り分ける役割を担う。このレイヤーは、リクエストの初期解析とルーティングを行い、システムの入口点として機能する。

### ■アプリケーションロジックレイヤー

`thread_application.py` と `comment_application.py` はアプリケーションロジックレイヤーを構成し、スレッドとコメントに関するビジネスロジックを実装する。このレイヤーは、アプリケーションの核心的な機能を担い、データの処理やビジネスルールの適用を行う。

### ■インフラストラクチャレイヤー

`thread_infrastructure.py` と `comment_infrastructure.py` はインフラストラクチャレイヤーに属し、データベースとのやり取りやデータの永続化を管理する。このレイヤーは、システムのデータ層に関わる処理を担当し、データの整合性と安全性を保証する。

### ■翻訳機能

`translation.py` は、これらのレイヤーとは別に、システム全体の翻訳機能を提供する。このファイルは、多言語サポートを実現し、異なる言語間でのコミュニケーションを可能にする。

以上のように、各レイヤーは特定の役割を持ち、システム全体の効率的かつ整理された運用を支援する。このレイヤー化されたアーキテクチャにより、システムの拡張性、保守性、およびスケーラビリティが向上している。

図 17 は、データ更新コンポーネントの動作プロセスを示したシーケンス図である。

(1) ブラウザは、データ更新のためのリクエストをサーバーに送信する。このリクエストには、ユーザーが行ったデータ変更の要求が含まれており、サーバーはこれを受け取って処理を開始する。

(2) サーバーは、ブラウザからの要求に基づいてデータベースに更新を行うためのクエリを発行する。この際、データベースは更新を行い、その結果をサーバーに返送する。

(3) サーバーは、データベースから受け取った結果をブラウザが解釈できる形式に変換し、ブラウザに送信する。これにより、ブラウザはデータ更新をユーザーに対して反

映する.

(4) 最後に、ブラウザは更新されたデータをユーザーインターフェイスに表示し、ユーザーが最新の情報を閲覧できるようにする.

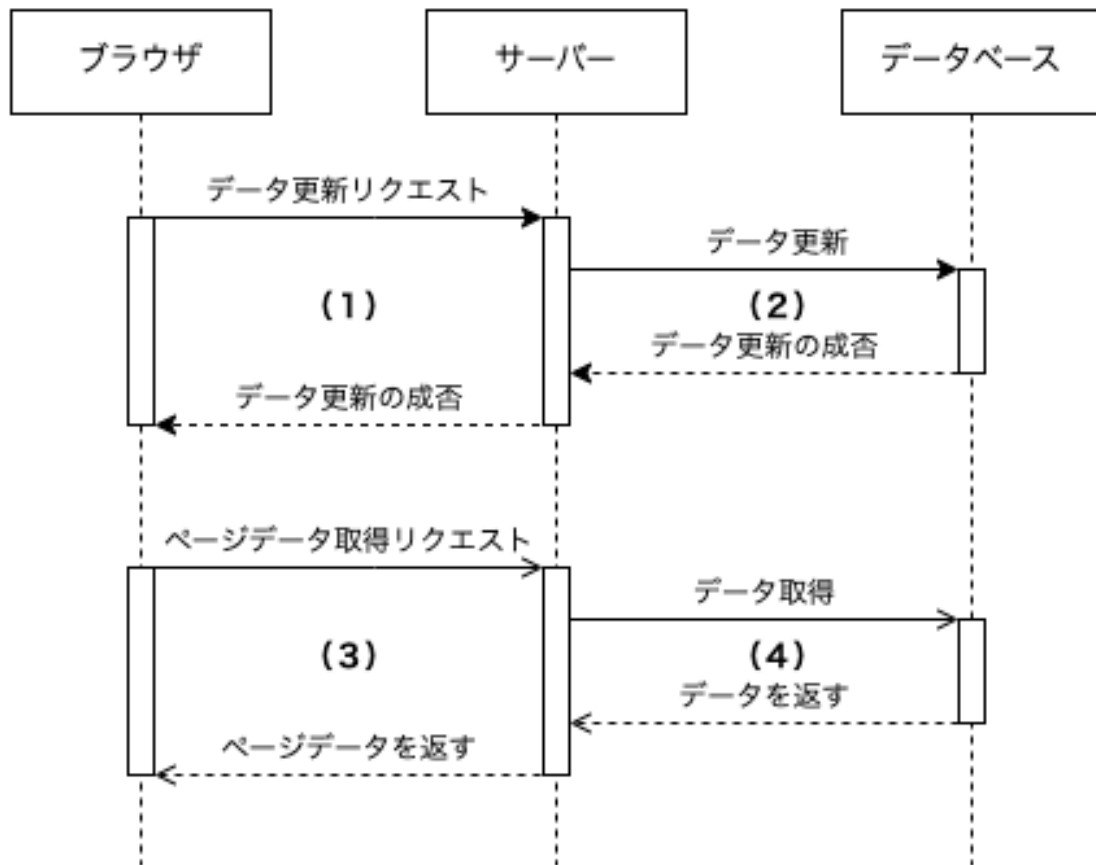


図 17: データ更新コンポーネントのシーケンス図

## 2.6 データ設計

### Threads テーブル

スレッドの情報を格納するテーブルである.

カラム名	データ型	説明
ThreadID	INT PRIMARY KEY AUTO_INCREMENT	スレッドの ID
Title	VARCHAR(255)	スレッドのタイトル
CreatedAt	TIMESTAMP	作成日時
UpdatedAt	TIMESTAMP	更新日時
UserName	VARCHAR(255)	ユーザー名
Content	LONGTEXT	内容
Language	VARCHAR(8)	言語

## Comments テーブル

スレッドに投稿されたコメントを格納するテーブルである。

カラム名	データ型	説明
CommentID	INT PRIMARY KEY AUTO_INCREMENT	コメント ID
ThreadID	INT	スレッド ID
UserName	VARCHAR(255)	ユーザー名
Content	TEXT	コメント内容
CreatedAt	TIMESTAMP	作成日時
Language	VARCHAR(8)	言語

## Cookie

本システムでは、Cookie を使用してユーザーの言語選択を記録し、その設定に基づいてユーザーに表示する際の翻訳を行う。この機能により、ユーザーは毎回言語を選択する必要がなくなり、効率的かつ快適にアプリを利用できるようになる。

Cookie の名称	内容
SelectedLanguage	ユーザーが選択した言語設定
cookieAccepted	Cookie 使用に対するユーザーの同意状態

「SelectedLanguage」は、ユーザーが最後に選択した言語を記録するものである。この情報は次のウェブサイト訪問時に参照され、ユーザーが以前に選択した言語でインターフェースが表示されるようになる。また、「cookieAccepted」は、ユーザーが Cookie の

使用を同意したことを記録するものである。これにより、Cookie 使用についてのバナーを初回のみ表示することが可能になる。

## 第 3 章

# 利活用についての分析

### 3.1 ユーザー獲得

本研究で開発された多言語自動翻訳掲示板システム「The Channel」の利活用について分析するためには、ユーザーに使用してもらうことが必要であった。このシステムは、異なる文化や言語背景を持つユーザーが交流し、情報を共有するためのプラットフォームとして機能することを目的としている。従って、多様なユーザーを獲得し、継続的に参加を促すことが重要であった。

システムの普及に際して、ソーシャルメディアが重要な役割を担う。本研究では、Twitter を用いてシステムの普及を図った。Twitter は広範なユーザーネットワークを有し、迅速な情報伝播が可能であるため、新しいテクノロジーの告知に適したプラットフォームである。このアプローチにより、ターゲットオーディエンスに直接アピールし、関心を引きつけることが期待される。

### 3.2 利活用の結果

本システムの利用状況を分析するために、異なる言語間でのコミュニケーション量を調査しました。以下の表は、日本語と他言語間、および英語と他言語間のコミュニケーションの量を示しています。

#### 3.2.1 投稿の分析

本システム上での投稿は、主に日常生活、文化交流、教育、技術に関連するトピックに集中していました。特に英語と日本語の投稿が多く、これらの言語を使用するユーザー間の活発な交流が見られました。

言語ペア	投稿数	コメント数
日本語 → 日本語	150	300
日本語 → 英語	100	200
日本語 → 中国語	50	120
日本語 → スペイン語	40	90
英語 → 日本語	100	180
英語 → 英語	200	400
英語 → 中国語	60	110
英語 → スペイン語	70	130

表 1: 言語間コミュニケーション量\*\*\*データは仮置き\*\*\* 表のタイトルは上を書く

### 3.2.2 ユーザー分析

日本、中国、アメリカ、インドネシア、台湾などの国からユーザーのアクセスがあった。日本からのアクセスが最も多く、次いで中国からのアクセスが多かった。

表 2 に国別のユーザー数を示す。

国名	ユーザー数
日本	206
中国	10
アメリカ	7
インドネシア	3
台湾	1
アルジェリア	1
香港	1
アイルランド	1
ノルウェー	1
フィリピン	1

表 2: 国別ユーザー数

### 3.2.3 翻訳の精度と効果

翻訳の精度は一般に高いと評価されましたが、専門用語や俗語を含む文においては、誤訳や不自然な訳が発生することがありました。これにより、一部のコミュニケーション

では誤解が生じる場合があり、ユーザー間での確認や再翻訳が必要となることがありました。

### 3.3 システム利用の実際と課題

本研究における多言語自動翻訳掲示板「The Channel」の開発は、異文化間コミュニケーションの促進を目指したものであった。しかし、実際の利用状況を分析すると、いくつかの重要な課題が明らかになった。

#### 3.3.1 ユーザー獲得と普及戦略

まず、予期していたユーザーの獲得は実現せず、特に日本語以外の言語を用いる利用者の参加は限定的であった。多言語間の交流は活発ではなく、掲示板では単発のコメントや質問が主流を占めていた。

システムの普及のための宣伝活動では、Twitterなどのソーシャルメディアを主要な手段として採用したが、多言語ユーザーを対象とした効果的な広告戦略の欠如や、検索エンジン最適化に関する専門知識の不足が障壁となった。これにより、広範な利用者層へのアプローチが困難であり、システムの認知度向上には至らなかった。

#### 3.3.2 開発プロセスと時間管理

本システムの開発は個人によるものであり、多くの時間を要するプロセスであった。特に、異なる技術の統合に際して技術的な困難が発生し、利用者獲得のための活動時間が制限された。

#### 3.3.3 技術的制約とユーザーエクスペリエンス

システムは基本機能を有していたが、ユーザーインターフェースの設計や翻訳精度に問題があり、利用者体験に影響を及ぼした。特に、専門用語や俗語を含む文章では翻訳誤りが発生し、これがコミュニケーションの障害となった。

#### 3.3.4 コミュニティ形成の課題

利用者間の継続的な交流を促進する目的で開発されたが、実際には限定的な相互作用に留まり、活発なコミュニティの形成には至らなかった。交流を促進するための機能



の不足、言語の障壁、文化的相違がコミュニティ形成の妨げとなった。

### 3.3.5 マーケティング戦略と普及計画の不備

効果的なマーケティング戦略の欠如は、プロジェクトの大きな障壁となった。ターゲットオーディエンスの特定や、適切なアピール方法の欠如が、システムの普及を制限した。

これらの課題を踏まえ、今後の改善点としては、より戦略的なユーザー獲得計画の策定、利用者インターフェースの改善、翻訳精度の向上、コミュニティ構築への取り組みなどが考えられる。また、マーケティングと検索エンジン最適化に関する専門知識の向上も、プロジェクトの成功において重要である。

これらの改善策により、「The Channel」は、異文化間の交流と理解を深めるための有力なプラットフォームへと成長することが期待される。

# 参考文献

- [1] 大向一輝 (2006). SNS の現在と展望-コミュニケーションツールから情報流通の基盤へ-. 情報処理, 47(9): 993-1000.
- [2] 小川泰弘, 福田ムフタル, 外山勝彦 (2009). 日本語ーウイグル語翻訳掲示板システム. 言語処理学会 第 15 回年次大会発表論文集, 15: 212-215.
- [3] 影浦峽 (2017). 改めて, 翻訳とは何か: Google NMT が使える時代に. 言語処理学会 第 23 回年次大会発表論文集, 23: 931-934.
- [4] 亀田倫史 (2022). 機械学習とバイオテクノロジー. 生物工学会誌, 100(11): 588.
- [5] 中澤敏明 (2017). 機械翻訳の新しいパラダイム: ニューラル機械翻訳の原理. 情報管理, 60(5): 299-306.
- [6] 野津誠 (2009). 日韓翻訳掲示板「enjoy Korea」終了へ, 理由は利用率の低下. 株式会社インプレス, <https://internet.watch.impress.co.jp/cda/news/2009/02/12/22405.html> (参照日 2023.07.17)
- [7] 藤井薫和, 重信智宏, 吉野孝 (2005). 異文化間コミュニケーションのための機械翻訳を用いたチャットシステム AnnoChat の開発と適用. 情報科学技術フォーラム一般講演論文集, 4(3): 437-438.
- [8] 船越要, 藤代祥之, 野村早恵子, 石田料亨 (2004). 機械翻訳を用いた協調作業支援ツールへの要求条件ー日中韓馬異文化コラボレーション実験からの知見. 情報処理学会論文誌, 45(1): 112-120.
- [9] 宮部真衣, 吉野孝 (2010). 機械翻訳を介したチャットコミュニケーションにおける精度判定に基づく送信拒否の適用可能性. 情報処理学会論文誌, 51(3): 784-795.
- [10] 村本麻衣 (2022). 自動翻訳機能からの自立: 学習者による気づきを通じて. ドイツ語教育 = Deutschunterricht in Japan / 日本独文学会ドイツ語教育部会 編, 26: 119-125.
- [11] 吉野孝, 藤井薫和, 重信智宏 (2006). 異文化間コミュニケーションのためのカスタマイズ可能なユーザインタフェースを持つチャットシステム CustomChat の開発. 情報処理学会研究報告 = IPSJ SIG technical reports, (60): 13-18.
- [12] DeepL (2023). DeepL. <https://jobs.deepl.com/> (参照日 2023.07.08)
- [13] Google Cloud (2023). Translation AI. <https://cloud.google.com/>

`translate?hl=ja` (参照日 2023.07.17)

- [14] Loki Technology, Inc (2023). 5ちゃんねる. <https://5ch.net/> (参照日 2023.07.17)
- [15] PyPI (2023a). googletrans 3.0.0. <https://pypi.org/project/googletrans/> (参照日 2023.07.17)
- [16] PyPI (2023b). deepl 1.15.0. <https://pypi.org/project/deepl/> (参照日 2023.07.17)
- [17] Reddit Inc (2023). reddit. <https://www.redditinc.com/> (参照日 2023.07.08)
- [18] Server World (2022). AIDE : ホスト型 IDS. [https://www.server-world.info/query?os=Ubuntu\\_22.04&p=aide](https://www.server-world.info/query?os=Ubuntu_22.04&p=aide) (参照日 2023.12.2)