

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	21
2.5	コンポーネントの詳細	21
2.6	データ設計	25
2.7	システム運用とユーザー獲得への取り組み	27
第 3 章	結果	28
3.1	利用状況の分析結果	28
第 4 章	考察	32
4.1	課題と改善策	32
4.2	総括と展望	33
	引用・参考文献	35

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, 2024) や DeepL API (DeepL, 2024a) がある。これらの API 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, 2023) や deepl-python (DeepL, 2024b) 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、それらは主に掲示板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掲示板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、この多言語自動翻訳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．この掲示板では，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．

掲示板システムの中核は，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．

この掲示板システムは，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．

2.2 主な機能と利用フロー

掲示板システム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。

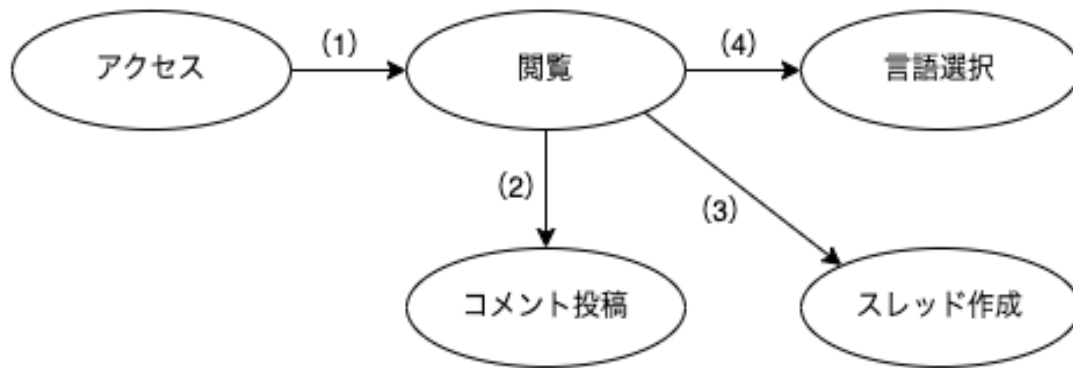


図 1. システム利用フロー

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。

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年だけここにいるので、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ができる、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.zauo.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 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

The screenshot shows a form titled '新規スレッド作成' (New Thread Creation). It contains three input fields: 'タイトル' (Title), '作成者名' (Creator Name), and '概要' (Summary). Each field has a red border and a red error message below it: 'タイトルは必須です' (Title is required), '作成者名は必須です' (Creator name is required), and '概要は必須です' (Summary is required). A blue button labeled 'スレッドを作成' (Create Thread) is located at the bottom right of the form.

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)。

The screenshot shows a form for posting a comment. It has two input fields: '作成者名' (Creator Name) and 'コメント' (Comment). A blue button labeled '送信' (Send) is located at the bottom right of the form.

図 7. コメント投稿欄

The screenshot shows the same comment posting form as in Figure 7, but with an error state. The 'コメント' (Comment) field is highlighted with a red border, and a red error message 'コメントは必須です' (Comment is required) is displayed below it. The '送信' (Send) button is still visible at the bottom right.

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。日ブというらしい。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

ユーザーはヘッダー (図 11) に設置された言語選択欄 (図 13) を通じて、システムの表示言語を自由に変更できる. この選択欄には言語がアルファベット順にリストされており、ユーザーは利用可能な言語の中から任意の言語を選択可能である. ここには、「Original」という特別な選択肢が用意されている. これを選択するとユーザーが入力した原文を翻訳せずにそのまま表示する. これにより、原文の内容をユーザーは確認することができる. また、「日本語」と「Original」は使用するユーザー数の多さを考慮してリストの先頭に配置されている.

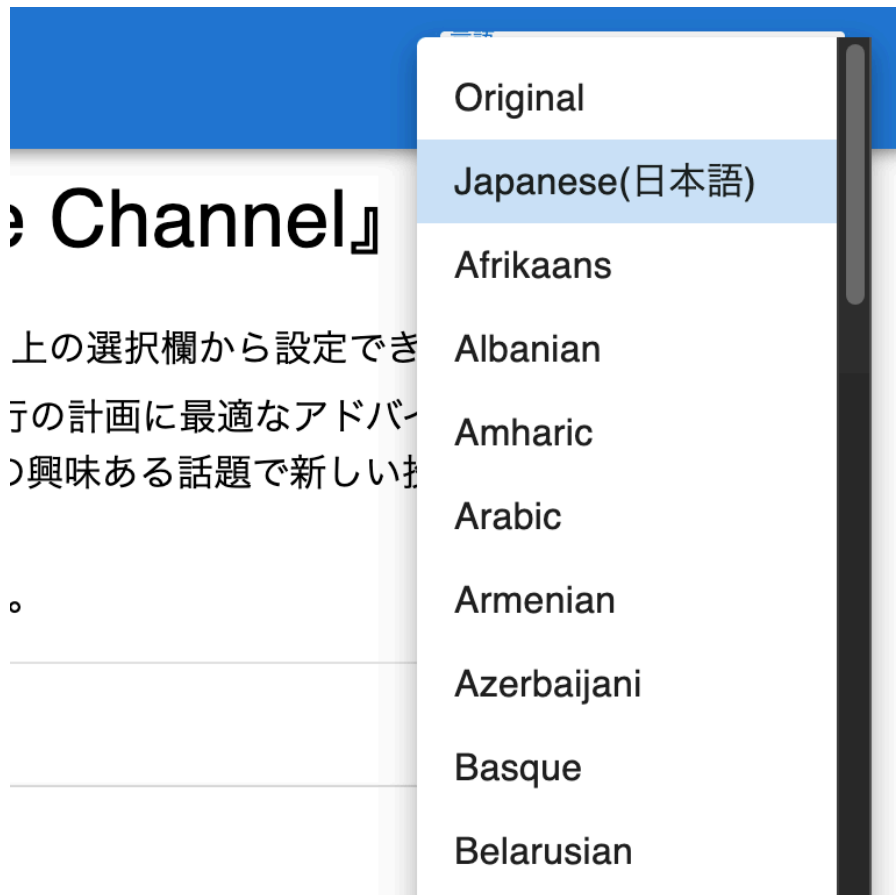


図 13. 言語選択欄

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 サーバーといった要素によって構成されている。図 15 にそれぞれの要素にお

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. 翻訳には Google Trans API を利用した.

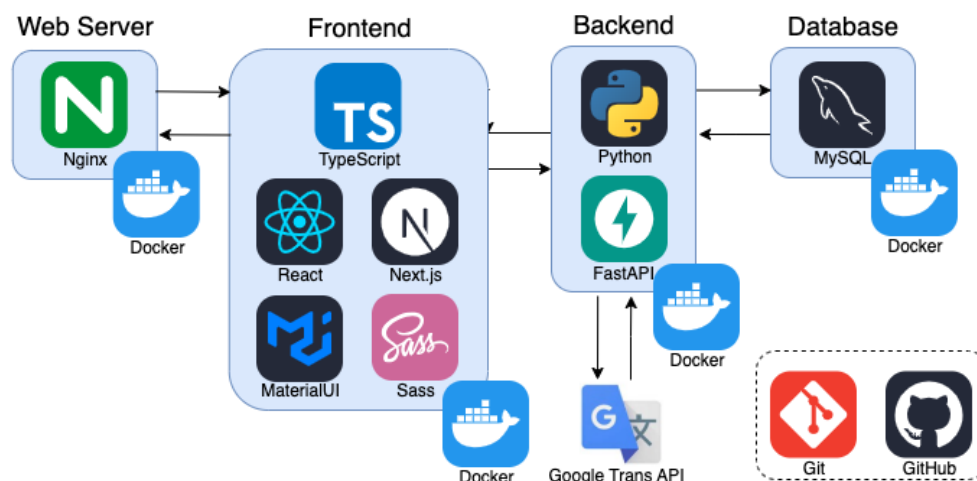


図 15. 技術スタック図

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

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

システムの保護を目的として、外部からの不正アクセスを遮断し、内部ネットワークのセキュリティを向上させ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

本研究のデータベースには, MySQL(8.0.1) を採用した.

2.3.6 通信の取り扱いと Nginx の役割

本研究で使用された Web サーバー Nginx の設定は, セキュリティを重視した基準に基づいて行われている. この設定は, 通信の安全性を確保しつつ, ユーザーのリクエストを効率的に処理することを目的としている.

通信プロトコルの変更

システムは公開初期には HTTP 通信を採用していたが, セキュリティの向上を目的として HTTPS 通信への移行を行った. この移行に伴い, Nginx には HTTP から HTTPS への自動リダイレクト設定が追加された. これにより, ユーザーが HTTP でアクセスした際には, 安全な HTTPS 通信へと自動的に転送されるようにしている.

プロキシとリバースプロキシの設定

Nginx は、フロントエンドとバックエンド間のリクエスト転送を効率化するプロキシおよびリバースプロキシサーバーとして機能している。具体的には、ユーザーがフロントエンドにアクセスする際に Nginx を経由し、必要なデータの取得を行う。また、ユーザーからの POST リクエストは Nginx を通じてバックエンドに転送され、処理される。

2.4 コンポーネント設計

システムは、表示コンポーネントおよ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ー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、ブラウザおよ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。

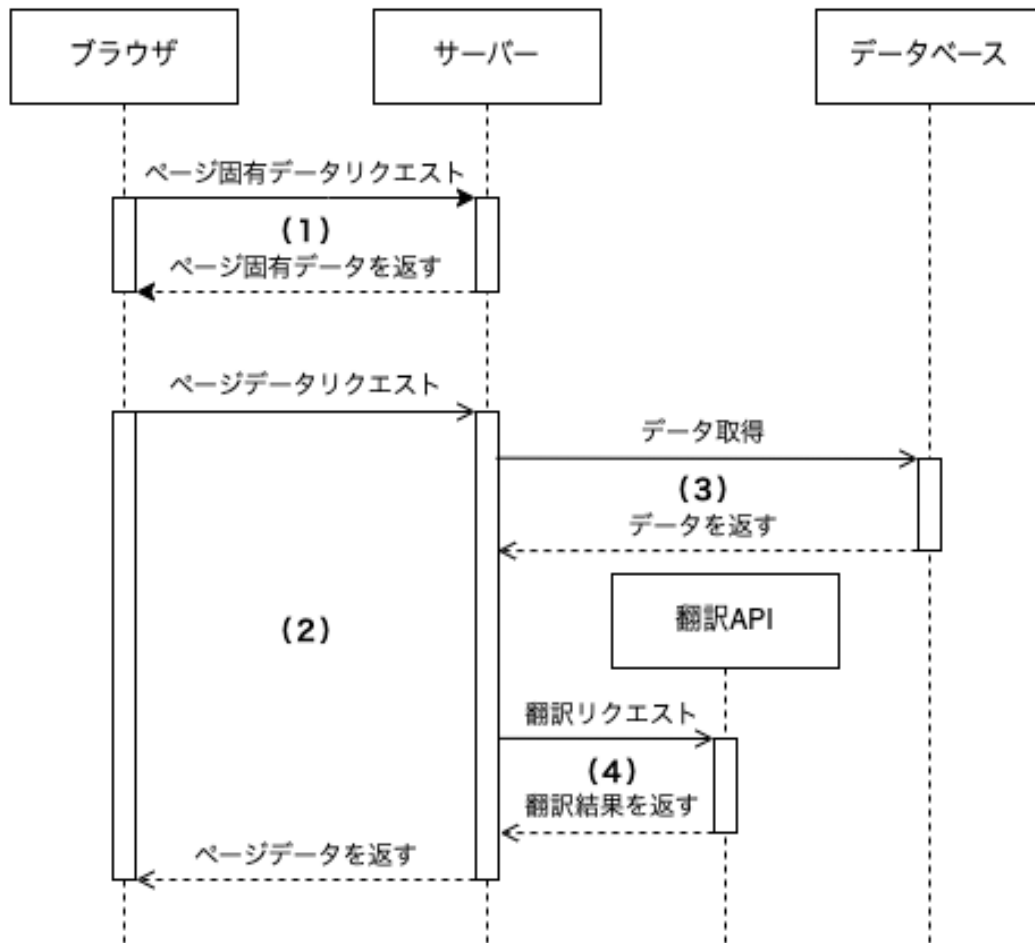


図 16. 表示コンポーネントのシーケンス図

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、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を向上させている。

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。

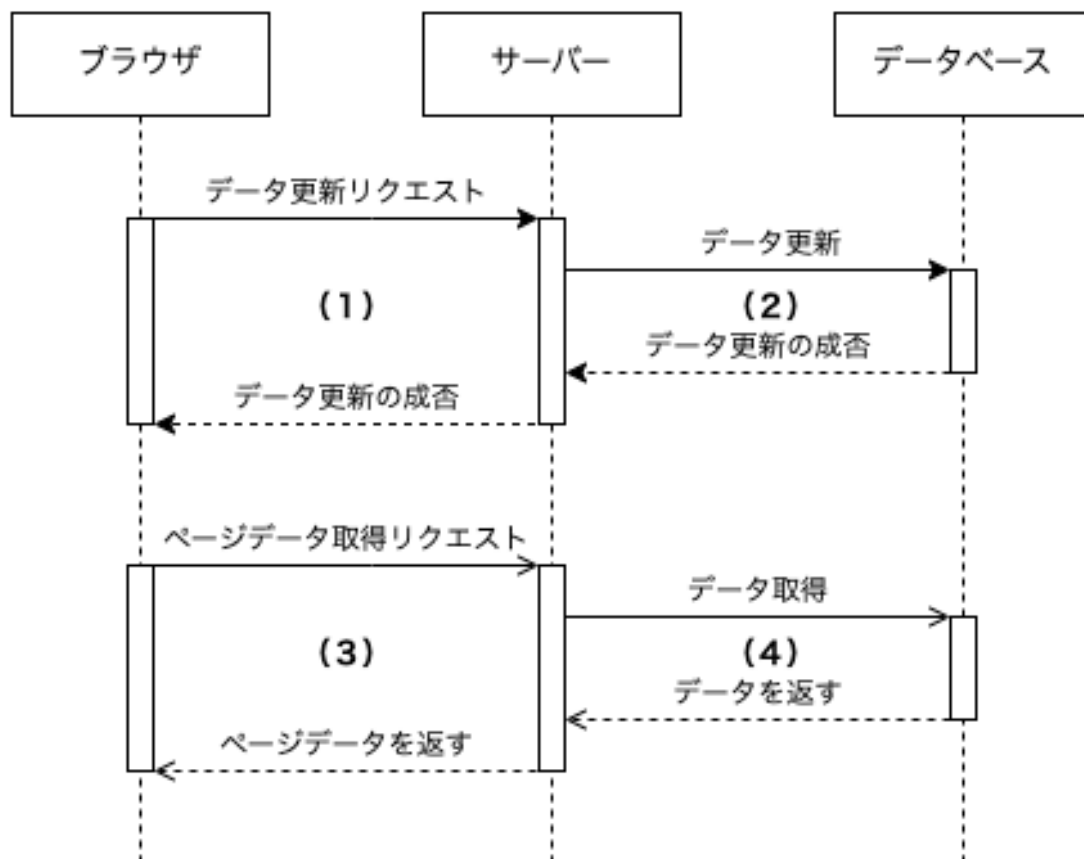


図 17. データ更新コンポーネントのシーケンス図

2.6 データ設計

スレッドテーブル (Threads)

スレッドの情報を格納するテーブルである。

表 1. スレッドテーブル

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

コメントテーブル (Comments)

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。

表 2. コメントテーブル

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 の使用について

本システムでは、ユーザー体験の向上を目的として 2 種類の Cookie を使用している。これらは以下の目的で利用されている。

■ 「SelectedLanguage」 言語設定の記録

最後に選択した言語設定を記録する。この情報は次回のウェブサイト訪問時に参照され、以前に選択された言語で画面が表示されるようになる。これにより、ユーザーは毎回言語を選択する手間を省くことができ、より効率的かつ快適にシステムを利用可能となる。

■「cookieAccepted」 Cookie 同意状態の記録

ユーザーが Cookie の使用に同意したか否かを記録する. これにより, Cookie の使用に関する通知バナーを初回訪問時のみ表示し, 再訪問時には表示しないようにすることが可能となる.

2.7 システム運用とユーザー獲得への取り組み

2023 年 10 月 17 日に, 多言語自動翻訳掲示板「The Channel」を Web 上に公開した. 公開後の 2023 年 11 月 2 日から, Google Analytics を利用したデータ収集とユーザー獲得に向けた活動を開始した. その際, Google Analytics を通じて, 訪問者の国別アクセス数や, システムへのアクセス方法などの情報を収集した.

ユーザー獲得に向けては, 以下の戦略を展開した:

1. **ソーシャルメディアを利用した宣伝活動**: Twitter を活用し, システムの普及を目指す宣伝活動を展開した. 掲示板の宣伝は, 日本語, 英語, 中国語, スペイン語, アラビア語, フランス語など複数言語で行った.
2. **口コミによる宣伝**: 筆者の友人や研究室のメンバーを通じて口コミでの宣伝を行った. 特に, 日本語以外のユーザーを対象として, 留学生や訪日外国人にアプローチした.
3. **ウェブ公開による流入促進**: インターネット上で掲示板を公開し, 全世界からのアクセスを可能にした.

第 3 章

結果

3.1 利用状況の分析結果

本システムの利用状況を分析するために、スレッドとコメントの内容や言語分布，国別のアクセスユーザー数，アクセス経路別のユーザー数，翻訳結果について分析した. 分析したデータは 2023 年 11 月 2 日から 2024 年 1 月 9 日の間に計測されたものである.

3.1.1 スレッドとコメント

表 3 にユーザーにより作成されたスレッドの言語分布を示す. スレッドの言語は googletrans (PyPI, 2023) を用いて特定した. 全体として 25 件のスレッドが作成され，その内容は食べ物，アニメ，日本に関するおすすめ，豆知識，災害情報など多岐にわたった.

表 3. 言語別スレッド数

言語	スレッド数
日本語	20
英語	3
フランス語	1
中国語	1

表 4 にコメントの言語分布を示す. これらも Google Translate API を用いて言語を特定した. URL のみの投稿は言語が特定できないため，表の数字には加算していない. 合計 70 件のコメントが投稿された. コメント内容はスレッドの内容に対する返答のようなものが大半を占めており，ユーザー間の対話はほぼ行われなかった.

表 4. 言語別コメント数

言語	コメント数
日本語	47
英語	7
中国語（簡体字）	4
中国語（繁体字）	4
アラビア語	3
フランス語	1
ヨルバ語	1
ペルシャ語	1
ズールー語	1

3.1.2 アクセスユーザー

表 5 に国別のアクセスユーザー数を示す. 日本, 中国, アメリカ, スペイン, インドネシアなどの国からアクセスがあった.

表 5. 国別アクセスユーザー数

国名	アクセスユーザー数
日本	224
中国	17
アメリカ	7
スペイン	4
インドネシア	3
香港	2
フィリピン	2
台湾	2
アルジェリア	1
エルサルバドル	1
アイルランド	1
ノルウェー	1

3.1.3 アクセス方法

表 6 にシステムへのアクセス方法ごとのユーザー数を示す。括弧内に記載されている用語は、Google Analytics におけるユーザーのアクセス経路を示す固有の表現である。

表 6. アクセス経路別ユーザー数

アクセス方法	ユーザー数
直接アクセス (Direct)	180
紹介 (Referral)	40
ソーシャルメディア経由 (Organic Social)	28
未割り当て (Unassigned)	17
検索経由 (Organic Search)	8

各アクセス方法の意味合いは以下の通りである。

- **直接アクセス (Direct)**：ユーザーが URL を直接入力してサイトにアクセスした場合。
- **紹介 (Referral)**：他のウェブサイトからのリンクを通じてサイトにアクセスした場合。
- **ソーシャルメディア経由 (Organic Social)**：ソーシャルメディアプラットフォームを経由してサイトにアクセスした場合。
- **未割り当て (Unassigned)**：アクセス方法が Google Analytics によって特定されなかった場合。
- **検索経由 (Organic Search)**：検索エンジンの結果を通じてサイトにアクセスした場合。

3.1.4 異言語間コミュニケーションの事例と翻訳誤差

翻訳機能を通じて、異なる言語間でのコミュニケーションが実際に行われたことが確認された。例として、”What are the best things to do in Tokyo?”というタイトルのスレッドでのやり取りを挙げる。ここでは、アメリカからの交換留学生が「1年間の日本滞在中に地元の人々が好むことを体験したい」という趣旨のスレッドを作成し、これに対して日本の様々なスポットや活動に関する多数のコメントが寄せられた。具体的には、魚釣りができるレストランの提案、駅内の立ち食いそば屋を巡る提案、日本の焼肉や山手線一周旅、水上バスやアニメイト訪問、貸切温泉の訪問などが挙げられた。これ

らのコメントは、翻訳機能を利用して異なる言語間での情報交換が行われたことを示している。

また、翻訳機能が意図しない言語認識を行う事例が観察された。一つの例として、日本語の文脈において身長を示す「3m」という表記が投稿された際、この表記が誤ってアラビア語と判定され、その結果、日本語への翻訳が「父方の叔父」と表示されるという事態が発生した。同様に、「Natto」という英語の単語が日本語に翻訳された際には、「ナット」と誤訳されるという事例も確認された。この場合、「Natto」という単語は文脈上「納豆」を指していたにも関わらず、翻訳昨日はこれを「ナット」と誤って解釈し、その結果として誤訳が生じた。

第 4 章

考察

4.1 課題と改善策

本研究で開発された多言語自動翻訳掲示板「The Channel」は、異文化間コミュニケーションの促進とそのコミュニケーションの分析を目的としていた。しかし、研究を通して掲示板の開発と、利用状況の分析をした結果、いくつかの課題が明らかになった。以下では、これらの課題を検討し、それぞれに対する具体的な改善策を提案する。

4.1.1 技術的課題

開発プロセス

本システムの実装は筆者一人によって行われた。加えて、筆者の Web アプリケーション開発に関する経験が影響を及ぼし、多大な時間が費やされた。その結果、利用者獲得に向けた活動に十分な時間を割り当てることが困難となり、普及戦略の展開に影響を与えた。この問題に対処するためには、十分な開発期間の確保、実装機能の選定、開発に関わる時間の適切な見積もりが必要である。掲示板システムに限らず、新たにシステムを開発する研究においては、これらの要素を考慮する必要がある。

翻訳

ユーザーが掲示板を使用した結果、翻訳機能が意図しない言語認識を行う事例が観察された。具体的には、表記された言語の誤認識により翻訳後の文章が不適切になる事例や、異なる言語間で正確な翻訳が行われない事例が発生した。これは、各コメントを個別に翻訳するプロセスが文脈を考慮せず行われた結果である可能性が高い。また、使用された翻訳技術自体の精度の不足も問題である可能性がある。これらの問題に対処するためには、文脈を考慮した翻訳が必要である。例えば、複数のコメントを一括で翻訳し、

翻訳機能に文脈を理解させる方法、あるいはユーザーが投稿する際に、その翻訳結果を例示し、ユーザーが意図した通りの翻訳が行われるかを確認する仕組みが考えられる。さらに、今回は無料で利用可能な翻訳 API を使用したが、有料で利用できるより高精度な翻訳 API の使用も有効な手段であると考えられる。実際に、最新の DeepL API (DeepL, 2024a) を利用して、翻訳機能が意図しない言語認識を行う事例の文章を翻訳したところ、正しく翻訳された。

4.1.2 ユーザー獲得と定着の課題

ユーザー獲得

予想されたユーザー数の獲得は達成されず、特に日本語以外を使用するユーザーの参加は限られていた。システム普及のための宣伝活動においては、Twitter や口コミによる宣伝を中心としたアプローチを採用した。しかし、多言語ユーザーに向けた有効な広告戦略の不足や、口コミによる宣伝の影響力不足が明らかになった。より多様な宣伝方法やターゲットを絞ったマーケティング戦略が必要である。

ユーザー定着

本研究において掲示板のインターフェースや機能は、日本の掲示板「5ちゃんねる」を参考に設計された。この設計は既存のユーザーにとっては馴染み深いものであったが、新規ユーザーに対しては、会話を促進する機能やインターフェースの工夫が不足していた。これが原因で、ユーザー間の継続的な対話やコミュニティ形成を促す要素が限定され、ユーザーの定着率に影響を及ぼした可能性がある。

また、掲示板を公開してからの運用期間の短さも影響を与えている可能性がある。「2ちゃんねるには2ちゃんねる語やアスキーアートによる定型的固有表現があるが、これを使うことによって2ちゃんねるの参加者は2ちゃんねるに参加していることを潜在的にせよ強く意識する」(松村ほか, 2004) ことが指摘されている。このような文化は長い時間をかけてユーザーの間で育まれていくものである。しかし、掲示板は公開から分析までの期間は3ヶ月ほどしかなかった。そのため、このようなユーザーが定着するための文化は生まれなかった。

4.2 総括と展望

本研究では、異なる文化や言語間のオンラインコミュニケーションを促進するために、「The Channel」という多言語自動翻訳掲示板を開発した。この掲示板は、異言語間の交流の可能性を探るものである。しかし、実際のユーザー獲得と定着には多くの課題が

あり，特に多言語ユーザーの参加と活発な交流の促進が期待に満たなかった．これは，新規ユーザーを引き付ける魅力的なコンテンツの不足や，効果的な宣伝戦略やコミュニティ文化の育成が不十分であったことに起因すると考えられる．今後は，技術面での翻訳性能の向上に加えて，効果的なマーケティング戦略やユーザー体験の向上により，多言語間コミュニケーションの促進を目指す．さらに，ユーザーからのデータを蓄積し分析することで，異なる言語や文化間の交流をより深く理解し，将来の研究に役立てることが期待される．

引用・参考文献

- DeepL (2023). DeepL. <https://jobs.deepl.com/> (参照日 2023.07.08)
- DeepL (2024a). DeepL API. <https://www.deepl.com/ja/docs-api> (参照日 2024.01.15)
- DeepL (2024b). deepl-python. <https://github.com/DeepLcom/deepl-python> (参照日 2024.01.17)
- 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.
- 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—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.
- Google Cloud (2024). Translation AI. <https://cloud.google.com/translate?hl=ja> (参照日 2024.01.17)
- 影浦峯 (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.
- 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.
- Loki Technology, Inc (2023). 5ちゃんねる. <https://5ch.net/> (参照日 2023.07.17)
- 松村真宏, 三浦麻子, 柴内康文, 大澤幸生, 石塚満 (2004). 2ちゃんねるが盛り上がるダイナミズム. 情報処理学会論文誌, 45(3): 1053-1061.
- 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.
- 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.
- 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.
- 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)

- 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.
- 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.
- PyPI (2023). googletrans 3.0.0. <https://pypi.org/project/googletrans/> (参照日 2023.07.17)
- Reddit Inc (2023). reddit. <https://www.redditinc.com/> (参照日 2023.07.08)
- Server World (2022). AIDE : ホスト型 IDS. https://www.server-world.info/query?os=Ubuntu_22.04&p=aide (参照日 2023.12.2)
- 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.