

**青山学院大学理工学部
情報テクノロジー学科
2024 年度卒業研究論文**

**研究題目
OCR とラベリングによる
書類整理自動化システムの構築と
有用性の評価**

2025 年 1 月 27 日提出

指導教員 Martin J. Dürst

15820094 : 嘉松 一汰

OCR とラベリングによる書類整理自動化システムの構築と有用性の評価

嘉松 一汰 (15820094)

Dürst 研究室

1 はじめに

ソフトウェアによって一連の事務処理を自動化する RPA (Robotics Process Automation) という技術は、企業で主に使用されるが、大企業と中小企業の間で 20% 以上の差があり、技術や規模による格差が顕著に現れている^[1]。これらの主な原因は、アナログ処理と RPA の親和性の低さ、企業の IT 知識の乏しさである。

本研究では、RPA を使用して紙媒体の処理を自動で実施するシステムを作成する。印刷された手書きの文字などを解読し、テキストデータに変換する OCR (Optional Character Recognition) 技術や、容易にシステムのバックエンドと連携可能なフレームワークである Ruby on Rails を使用する。技術革新に伴いデジタル化に遅れを取り、最新の技術に関する知見の少ない企業や、システム障害への恐怖感が原因で、IT に対してネガティブな印象を抱いている企業が一定数ある現状の改善を目指す。

2 提案手法

本研究で作成するシステムは、電子ファイルではなく紙媒体中心で業務を行っており、他社と比較して、システム基盤がデジタル化できていない企業をターゲットにしている。中でも、山積みになった書類を職種ごとに分類する業務に、日々多くの時間を割いている企業に対して有効である。本システムを活用することで、手間のかかる業務を省くことができるため、生産性向上や社員のストレス軽減が見込まれる。以下にその具体的な処理の流れを示す。

まず、カテゴリ化したいドキュメントファイルに対して Google Vision API を使用した OCR を実施し、内容の単語群を取得する^[2]。その後、日本語、英語に対応した大規模データベースである WordNet と連携し、単語ごとの定義を取得する^[3]。各々の定義を元に単語がどのカテゴリに属するかを判別し、単語ごとにラベルを付与する。結果から、最終的なドキュメントのカテゴリを決定する。なお、各々のカテゴリに関しては、WordNet の Synset モデルにカテゴリ名を入力し、単語群を取得する方法で用意した。

2.1 使用する WordNet のテーブル

本研究で使用する WordNet テーブルには 3 種類あり、図 1 にこれらの ER 図を示す

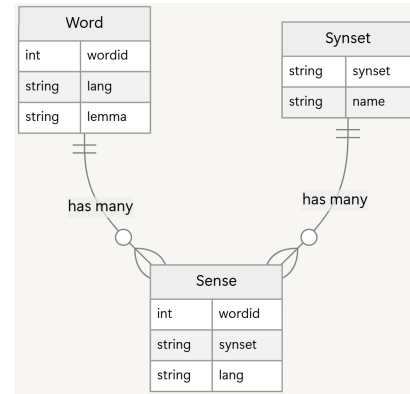


図 1: WordNet のテーブル構成

2.2 モデル設計

本研究で使用するモデルは 3 種類ある。WordNet のテーブルと同名で作成し、単語や定義の取得に必要なカラムのみを抽出している。

Word モデルには、単語に関する情報が格納されており、主キーは整数型の wordid カラムである。Sense モデルと一对多の関係で、複数の Sense モデルへのリレーションを持っており、Sense モデルを経由して、Synset モデルへのリレーションを辿ることで、単語からその定義を取得する。場合に応じて適切な言語のレコードを活用するため、言語での絞り込みをするスコープを持つ。

Synset モデルには、単語の定義に関する情報が格納されており、主キーは文字列型の synset カラムである。Sense モデルと一对多の関係で、複数の Word モデルへのリレーションを持っており、Sense モデルを経由して、Word モデルへのリレーションを辿ることで、定義をもつ単語を取得する。Word モデルと同様に、言語での絞り込みをするスコープを持つ。

Sense モデルは、上記の 2 つのモデルそれぞれと一对多のリレーションを持っている。synset カラムと wordid カラムが格納されており、それらを外部キーとして使用している。カラムと 2 つのモデルの双方向に紐づける役割を担う。単語と定義間で実行

される処理は、Sense モデルを経由する。

3 実験・評価

本研究の実験で用意したカテゴリは、経理、人事、庶務の3種類であり、各カテゴリごとに5個、計15個のドキュメントに対してカテゴリ分けし、精度分析を実施した。結果として、処理精度にはドキュメントやカテゴリごとにばらつきが見られた。特に、旧字体を含む文書では処理が失敗する傾向が確認された。

一方で、縦書きや横書きが混在する文書、英語表記が一部含まれる文書、ゴシック体や太文字を用いた文書などにおいては、問題なく処理が成功した。これらの結果から、フォーマットや字体の違いなどの、内容の構成によらず精度は担保されている。また、カテゴリごとの処理精度について、経理カテゴリにおける精度が他のカテゴリと比べて著しく低いことが確認された。この主な要因としては、経理カテゴリの文書に旧字体や、特定の業界でしか使用されていない単語が多く含まれている点が挙げられる。

4 考察

本システムの優位点については、ドキュメントの形式に関わらず同様の処理結果を得ることができる点である。文書には、縦書きや横書き及びそれらの複合など、内容の形式はジャンルによって多岐にわたる。それら全てに対応できなければ、システムの優位性は著しく低下する。しかし、さまざまな形式の文書を実験では用意したが、特に形式の差による精度の変化は見受けられなかった。そのため、システムに対応可能な形式にドキュメントを修正する必要がなく、その項目の有用性は担保されている。紙媒体中心の業務に不便を感じている人が現代に多くいる現状を改善するという観点では、要件を満たすことができているといえるだろう。

しかし、改善点も見受けられた。OCR 処理では不自由なく単語ごとに分けることができているが、WordNet データベースにその単語が存在しなかった場合、単語のカテゴリが判別できない点である。言語の観点からも同様の考察ができる。WordNet は日本語と英語を主な対象としているが、グローバル化が進む現代社会では、多言語対応が求められる場面が増えている。そのため、現在使用している WordNet がないデータセットを扱っているデータベースや、多言語に特化した語彙データベースを統合し、判別可能な語彙を拡張する必要がある。

5 おわりに

本システムは、現在の業務フローにおける課題を解決し得る有効なツールであると結論付けられる。今後も IT 分野の著しい成長に遅れをとった企業を中

心に、さまざまな業務的ニーズが発生すると考えられる。本研究では紙媒体中心業務をさらに細分化した、ドキュメントのカテゴライズ分野に着目してシステムの作成をしたが、本システムのようなソリューションが生み出されることで、IT 社会の課題が解決されることを願っている。

今後の展望として、ユーザーインターフェース (UI) の最適化も重要である。現状ではシステムのコア機能に重点を置いているが、操作の直感性やユーザーエクスペリエンスの向上が求められる。例えば、分析結果を可視化するダッシュボードや、フィードバック機能を備えたインターフェースを実装することで、利用者の利便性を高めることが可能である。また、システムをクラウドベースで提供することにより、複数のユーザーが同時に利用できる環境を構築し、スケーラビリティを確保することができる。

さらに、API を公開することで、他のアプリケーションやサービスとの統合を可能にし、システムの拡張性をさらに高めることも有効である。

参考文献

- [1] 佐々木康浩. RPA (Robotic Process Automation) の可能性. 経営情報学会 全国研究発表大会要旨集, Vol. 2017s, pp. 201–204, 2017.
- [2] 大西 淑雅. 文字認識 API を用いた講義アーカイブシステムの設計. 第 2018s 巻, pp. 1–6, 2018.
- [3] 竹内孔一. 日本語 Wordnet における語義・概念の分散表現獲得. 第 18 巻, pp. 245–246, 2019.

OCR とラベリングによる
書類整理自動化システムの構築と
有用性の評価

青山学院大学理工学部
情報テクノロジー学科Dürst研究室

学生番号：15820094

嘉松 一汰

目次

| | | |
|------------|--|----------|
| 第1章 | はじめに | 1 |
| 1.1 | 研究背景 | 1 |
| 1.2 | 研究目的 | 2 |
| 1.3 | 論文の構成 | 3 |
| 第2章 | 関連技術 | 4 |
| 2.1 | 基礎技術 | 4 |
| 2.1.1 | Ruby | 4 |
| 2.1.2 | Ruby on Rails | 4 |
| 2.1.3 | OCR (Optional Character Recognition) | 4 |
| 2.1.4 | Google Cloud Vision API | 5 |
| 2.1.5 | WordNet | 5 |
| 2.2 | クラスタリングの既存手法 | 5 |
| 2.2.1 | 非階層的クラスタリング | 5 |
| 2.2.2 | 階層的クラスタリング | 5 |
| 第3章 | 関連研究 | 6 |
| 3.1 | RPA に関する研究 | 6 |
| 3.1.1 | RPA の歴史 | 6 |
| 3.1.2 | RPA の将来と可能性 | 6 |
| 3.2 | カテゴリズに関する先行研究 | 7 |
| 3.2.1 | 機械学習を用いたカテゴリズ | 7 |
| 3.2.2 | ドキュメントのカテゴリズ | 7 |
| 3.3 | 先行研究との比較 | 8 |
| 3.3.1 | 先行研究との類似点 | 8 |
| 3.3.2 | 先行研究との差異 | 8 |
| 第4章 | 提案手法 | 9 |
| 4.1 | システム設計 | 9 |
| 4.1.1 | データベース設計 | 9 |
| 4.1.2 | モデル設計 | 10 |
| 4.2 | 提案する手法・アプローチ | 11 |

| | | |
|--------------|--------------------------------|-----------|
| 4.2.1 | ドキュメントファイルをアップロード | 11 |
| 4.2.2 | OCR によってドキュメントの内容を取得 | 12 |
| 4.2.3 | 旧字体を新字体に変換 | 13 |
| 4.2.4 | それぞれの単語の定義の取得 | 14 |
| 4.2.5 | 単語ごとにカテゴライズ, ラベリング | 15 |
| 4.2.6 | ラベリング結果から文章のカテゴリを決定 | 16 |
| 第 5 章 | 実験・評価 | 17 |
| 5.1 | 実験で使⽤したカテゴリ・ドキュメント | 17 |
| 5.2 | 精度の分析 | 18 |
| 5.2.1 | タイトルごとの精度 | 18 |
| 5.2.2 | カテゴリごとの精度 | 20 |
| 第 6 章 | 考察 | 22 |
| 6.1 | ドキュメントごとの結果の解釈 | 22 |
| 6.2 | 改善点 | 22 |
| 第 7 章 | おわりに | 24 |
| 7.1 | 研究のまとめ | 24 |
| 7.2 | 今後の展望 | 24 |
| | 謝辞 | 25 |
| | 参考文献 | 26 |
| | 付録 | 28 |
| | 付録 A | |
| | 質疑応答 | A-1 |
| | 付録 B | |
| | 実験で使⽤したドキュメント | B-1 |
| | 付録 C | |
| | カテゴライズの判別基準となる単語群 | C-1 |

第1章 はじめに

1.1 研究背景

日本企業の RPA (Robotics Process Automation [1]) 導入率は全体で 38%, 中小企業では 25% となっており, 非常に少ないことがわかる (図 1.1). また, 大企業と中小企業の間に 20% 以上の差があり, 技術や規模による格差も見えて取れる. これらの原因となっており要因として考えられることは, 大きく分けて 2 つある.

1 つ目は, RPA には専門領域と非専門領域が存在するということである. 専門領域は PC 上の操作や, デジタルの領域における処理である. いっぽうで, 紙媒体の処理等の, アナログの世界で実施される処理は非専門領域としている. 特に, 手書きの文字や画像の認識を高い精度を保ちながら自動で処理することは, 現代では非常に困難なことである. 具体的には, 縦書き文字横書き文字が混在していたり, 旧字体や特殊文字等の組み合わせも考えられるため, 例外的な処理までを自動でする必要があるからである [2].

2 つ目の原因は, 紙媒体の業務を実施している企業の IT 知識の乏しさにある. 詳しくは次のセクションで説明する. このような現状を踏まえて, 次章からは, OCR 技術を使用したアプローチを提案する.

また, RPA の定義についても, 明確には定まっていない部分が多い. 自動化という概念自体に対して RPA という言葉を適用するとしたら, 工場で食品の生産や梱包をするロボットも, 我々の家で活躍するお掃除ロボットも, PC 上のボタン一つで複数の処理をするシステムも全て RPA と呼ぶことができる. そのため, 本研究では, RPA という言葉の意味を広義的に捉え, RPA の非専門領域で実施される処理を, 他の様々な技術を用いて克服することを目指す. より詳細な技術については第 2 章で述べるが, OCR や 字句解析の技術を使用し, 紙媒体に対する処理を不自由なく自動化し, Ruby on Rails を使用したシステムとしての運用をして, 我々の周りに多くある Web アプリケーションと同様に利用できるようにすることで, IT に関する知識が乏しい企業でも, 安全かつ快適に自動化の恩恵を受けることを目指す.



図 1.1: 企業の RPA 導入率

1.2 研究目的

本研究の目的は、IT 知識が乏しく、紙媒体の業務を実施している企業を中心に、RPA を使用して紙媒体の処理を自動で実施するシステムを作成することである。具体的には、紙媒体の処理を OCR (Optical Character Recognition) 技術を用いてデジタルデータに変換し、さらに文字認識結果を用いた自動分類・ラベリングによって、手作業の削減と業務効率の向上を目指す。

日本で働く人事・総務担当者に、「紙媒体中心の業務で不便を感じたことがあるか？」とアンケートを取ったところ、61%が不便を感じたことがあると回答した(図 1.2)。上記の理由として、システム障害への恐怖感や、IT 知識の乏しさが挙げられる。多くの企業がデジタル化に対する適応を進めている中、依然として紙媒体を中心とした業務フローに頼らざるを得ない状況がある。これにより、文書の管理や検索に時間がかかるだけでなく、人的ミスや紛失のリスクも存在する。

また、既存の RPA ソリューションでは紙媒体の取り扱いが難しく、専用の高価な機器が必要となる場合がある。本研究では、これらの課題を克服し、誰でも簡単に使用できるシステムを設計・実装する。紙ベースの情報管理を効率化し、日常業務で活用できるようにする。

また、電子帳簿保存法の改正により [3]、企業は税務関連の書類や請求書などを電子データとして保存することが法的に求められるようになった。この法律が施行されたことで、企業に対してある程度のデジタル化が義務付けられたことになる。本研究のシステムを導入することで、電子帳簿保存法に適合したデータ管理の効率化が期待できる。これにより、企業のコンプライアンス遵守を支援し、業務負担の削減を目指す。

さらに、システムを Web アプリケーションとして提供し、スマートフォンやタブレットからも簡単にアクセス可能な形にすることで、利便性向上を目指す。このような RPA ソリューションを普及させることにより、デジタル化が遅れている分野でも手軽に導入できる環境を提供し、業務の自動化を促進することを目指す。

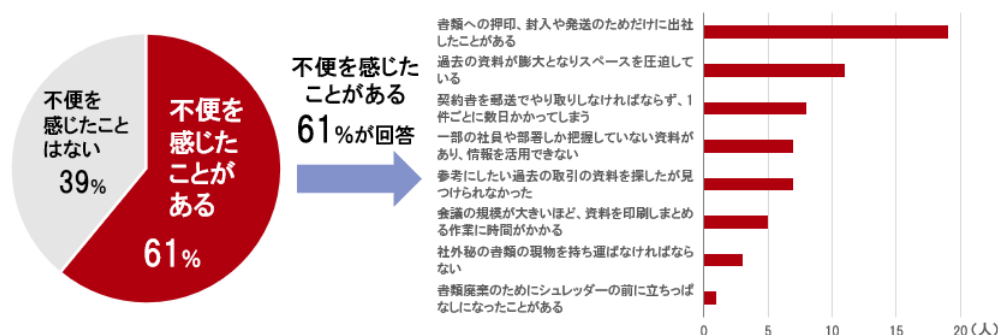


図 1.2: 紙媒体中心業務に関するアンケート

1.3 論文の構成

本論文は7章構成になっている。第1章では、本研究の背景や目的を述べた。第2章では本研究で使用した基礎技術、関連技術を述べる。第3章では、先行研究について説明し、本研究との差異を述べる。本研究での実装手法を第4章で提案し、第5章で実験と評価、第6章で結果の考察を述べ、第7章でに本研究の総括をする。

第2章 関連技術

2.1 基礎技術

2.1.1 Ruby

Ruby [4] は、1993 年にまつもとゆきひろ (松本行弘, 通称 Matz) が日本で開発したオブジェクト指向言語である。動的言語であり、変更への強さやバグチェックの容易さなど、シンプルな文法で構成されている [5]。名前はまつもとの同僚の誕生石であるルビーが由来となっている。本研究では、システムのバックエンドでのテーブル操作に使用する。

2.1.2 Ruby on Rails

デンマークのプログラマであるデイヴィッド・ハイネマイヤー・ハンソン (通称 DHH) 氏によって作られた。エンジニアの間では略称である Rails または RoR と呼ばれることも多く、簡単なコードで Web アプリケーションの開発ができるように設計されている [6]。本研究では、フロントエンドの ERB ファイルを始めとした、Web システムとしての実装に使用する。

2.1.3 OCR (Optional Character Recognition)

印刷された文字や手書きの文字などをカメラやスキャナ等の光学的な手段でデータとして取り込み、それを解読 (文字認識) することによって一度印刷されてしまった文字をパソコンなどのコンピューターが利用できる文字 (テキスト) データに変換する技術。データ入力作業の手間を大幅に削減し 2 重入力や人的ミスの削減などを目的とした利用はビジネス用途にも広く浸透しており、流通・製造・医療・小売などあらゆる業界で本来は人が読むために印刷された文字をコンピューターに取り込みたいという要望が根強くあり、バーコードや 2 次元コードが普及した現在でも需要はむしろ高まる傾向にある。

2.1.4 Google Cloud Vision API

Google が提供する画像認識サービスのことである。Google 独自の機械学習モデルを採用しており、効率的に画像を分析し、オブジェクトや顔の検出や手書き文字の読み取り、有用な画像メタデータの構築など様々なことを実現できる [7]。本研究では、ドキュメントの画像に対して OCR 処理をして、単語ごとに分類するために使用する。

2.1.5 WordNet

大規模な語彙データベースのこと。名詞、動詞、形容詞、副詞は、認知同義語 (Synset) のセットにグループ化され、それぞれが異なる概念を表現する。概念は、概念的意味的および語彙的關係によって相互にリンクされている。テーブル構造としては、概念テーブル (Synset)、語彙テーブル (Word) とそのリレーションテーブル (Sense) によって形成されている [8]。本研究では、単語を該当するカテゴリに分け、ラベリングをするために使用する。

2.2 クラスタリングの既存手法

本研究では、あくまでカテゴリライズという目的でシステムを構築しているため、クラスタリング手法とは少し異なる点があるが、アルゴリズム等の観点で参考にしたため、関連技術として以下に示す。

2.2.1 非階層的クラスタリング

目的のデータを事前に定義されたクラスタ数に分解することによって処理されるクラスタリング方式のことである。代表的な手法として、クラスタ内の分散を最小化するようにデータポイントをグループ化する k-means アルゴリズムが挙げられる [9]。

2.2.2 階層的クラスタリング

データポイントを機構造の階層に分割して処理するクラスタリング方式のことである。代表的な手法には大きく分けて、凝集型と分割型の2種類がある。凝集型は、木構造の下から上へクラスタを統合していく方法である。対して分割型は、上から下へクラスタを分割していく方法である。データの類似度に基づいて徐々にクラスタを形成していくため、クラスタ数を事前に決定する必要がなく、クラスタの適切な数が不明瞭な場合や、データの階層的な構造を理解したい場合には分割型が有用である [10]。

第3章 関連研究

3.1 RPA に関する研究

3.1.1 RPA の歴史

Madakam ら [11] は, RPA の技術的発展とその適用範囲について調査している. RPA がどのように進化してきたか, また, どのような業務プロセスに適用されてきたかについて述べている.

2010 年代前半の RPA は, 単純な定型業務の自動化を目的として開発され, そのシンプルな仕組みが特徴であった. しかし, 2018 年に本格的な業務への展開が加速し, 実際に金融, ヘルスケア, 製造業, 行政機関など, さまざまな業界で採用が広がり, year of Robotics Process Automation (RPA の年) と称されるようになった. また, 近年では AI (人工知能) や機械学習との統合により, より高度な業務への適用が可能になっている. そのため, 企業が競争力を維持するために RPA 導入を急速に進めていることが示されている.

特に, RPA の適用事例として, 人事業務, 経理処理, 請求書管理, データ移行などのバックオフィス業務における導入効果を詳細に分析している. これらの業務は, 本研究で開発するシステムの適用対象とも重なるため, 本研究の有用性を裏付ける重要な知見となる.

3.1.2 RPA の将来と可能性

Lee ら [12] は, RPA が雇用に与える影響について詳細に論じている. 彼らの研究では, RPA が企業の業務効率化やコスト削減に寄与する一方で, 人間の雇用に対する影響についても議論されている. 具体的には, RPA の導入が進むことで, 反復的なルーチンタスクを自動化し, 生産性の向上やヒューマンエラーの削減, 24 時間 365 日の稼働等のメリットをもたらすと述べられている. 一方で, RPA の普及により, 一部の業務が自動化されることで雇用の減少につながる可能性も指摘されている.

また、RPA が最も影響を与える業界や職種を特定し、企業の適応戦略についても分析している。特に、RPA の導入によって、ホワイトカラー業務のあり方が変化していくことを強調している。これらの知見は、RPA の導入に対して企業がどのように向き合うべきかを示唆するものであり、本研究の目的と関連性が高い。

3.2 カテゴリズに関する先行研究

3.2.1 機械学習を用いたカテゴリズ

花房ら [13] は、プログラミング読解中の視線軌道を機械学習によりカテゴリズすることを試みた。従来、プログラミング技能の評価は主観的な評価や定性的な分析が中心であった。しかし、視線追跡データを用い、教師あり学習を用いた学習モデルの 1 つである SVM (Support Vector Machine) を活用することにより、技能の異なる学習者（得意群・普通群・不得意群）の視線軌道パターンの定量的に分類した。大学 4 年生 24 名を対象とし、C 言語のプログラムを提示しながら視線の動きを計測した。ヒートマップデータを簡略化し、視線パターンの特徴を抽出した。

任意のデータを定量的に分類し、精度を分析している点は、本研究と共通している。主観的な評価に依存せず客観的な分析を実現している点は、機械学習によるカテゴリズの利点といえるだろう。一方で、低難易度の問題では分類精度が低下したという結果から、解析結果の信頼性が、使用するデータセットに依存してしまう点は、カテゴリズの課題点だといえる。

3.2.2 ドキュメントのカテゴリズ

松田ら [14] は、Web ページ (HTML 文書) を、HTML タグと、その要素の情報によってカテゴリズすることで、WWW 検索システムの拡張を試みた。従来の検索の仕組みは、検索キーワードと全てのページを照合し、一致率の高い順番で表示する。しかし、あらかじめ、Web ページをいくつかのカテゴリに分類しておくことで、検索速度や、内容一致度の向上が見込まれる。

処理の具体的な流れを示す。カテゴリズの基準となる特徴記述ファイルを用意する。ファイル内には、タグと要素の組み合わせごとに、どのカテゴリに分類されるかの詳細な条件が記述されている。読み込んだ HTML ファイルと特徴記述ファイルを比較し、文書のカテゴリを決定する。

文章に対して内容を解析し、カテゴリズするという点が本研究と共通している。特に解析方法について、Web ページ全体を複数の要素に分割し、各々に対して処理をした後に、文書全体のカテゴリを判別するという流れは、本研究でも取り入れている。

3.3 先行研究との比較

3.3.1 先行研究との類似点

第 3.2 節で示した 2 つの先行研究と同様に、特定の媒体に対してカテゴリズを実施しており、主観的な判断を排除しデータに基づく定量的な結果の分析を重視している。特に松田らが実施した、カテゴリズに関する研究 [14] では、文書を要素ごとに分割し、それぞれのカテゴリを判別する手法を採用しているのに対し、本研究でも文書を単語レベルに分割し、各単語の定義を解析することで文書全体のカテゴリを決定するアプローチを採用している。

3.3.2 先行研究との差異

一方で、これらの研究との大きな違いは、システムのコストパフォーマンスと拡張性である。花房らが実施した、視線誘導に関する研究 [13] では、機械学習モデルを使用しているが、このような高い精度を担保するモデルを 1 つの処理ごとに使用すると、時間的、金銭的にも高コストになってしまう。本研究では、そのような機械学習モデルは使用しておらず、少ないリソースで構築可能なシステムになっている。さらに、拡張性に関しても、機械学習モデルを使用している場合はそのモデルにある程度依存した結果しか得られないが、本研究ではカテゴリズの条件を自由にカスタマイズできるため、各々の企業に最適なシステムを追求することができる。

また、松田らが実施した、文書タイプのカテゴリズに関する研究 [14] では、自身でカテゴリズの基準となるファイルを作成する必要がある。カテゴリが増えるたびに判別条件を考える必要がある。本研究では、単語の定義からカテゴリズするため、カテゴリに該当する単語を容易に取得することができ、システムの構築や拡張段階において、コストを低く抑えることができる。

第4章 提案手法

本研究で活用する手法，それに対するアプローチをはじめ，テーブルの構造や内容などのシステム設計を示す．また，WordNet データベースとの連携方法に加えて，ドキュメントをどのようにカテゴライズするかを具体的に示す．

4.1 システム設計

本研究のシステムのバックエンドに該当する部分を具体的に示す．

4.1.1 データベース設計

本研究で使用するテーブルには3種類ある．図 4.1 に，これらを表す ER 図を示す．

Word テーブル Wordnet データベースの単語が格納されており，主キーである整数型の wordid カラム，言語を表す文字列型の lang カラム，単語名を表す文字列型の lemma カラムによって形成されている．

Synset テーブル WordNet データベースに含まれる単語の定義が格納されており，主キーである文字列型の synset カラム 定義名を表す name カラムによって形成されている．

Sense テーブル 前述した2つのテーブルの中間テーブルであり，Word テーブルとの外部キーである wordid カラム，Synset テーブルとの外部キーである synset カラム，言語を表す文字列型の lang カラムによって形成されている．

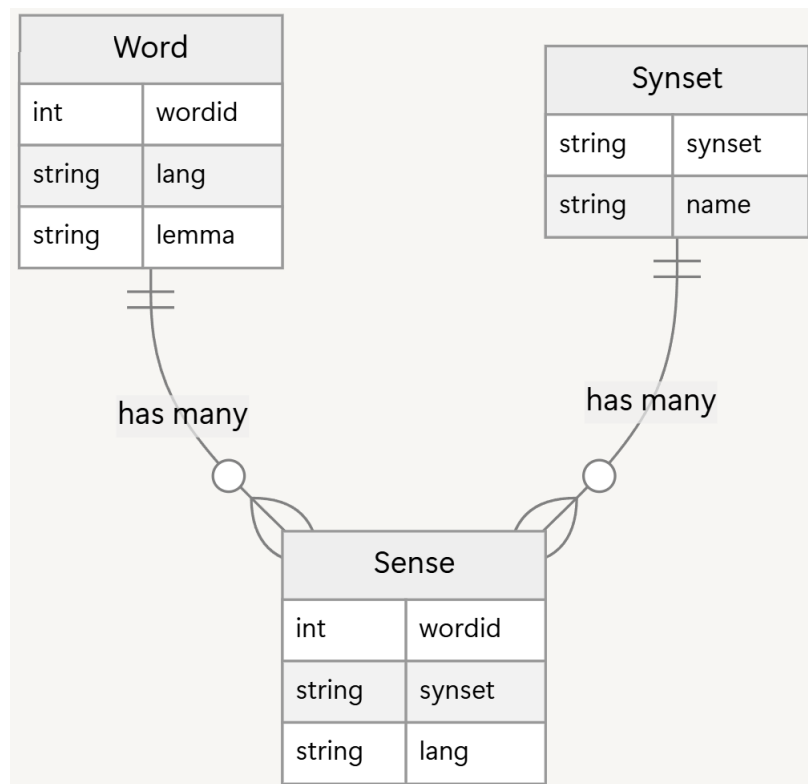


図 4.1: データベースの ER 図

4.1.2 モデル設計

Ruby on Rails にはモデルという概念があり，データベースの内容をモデルとして定義することで，CRUD 処理を始めとしたバックエンドでのデータベースのやり取りを効率的にすることができる．本研究では第 4.1.1 節で示したテーブルをモデルとして定義し，整合性等のデータベースレベルの具体的なプロパティを設定する．

Word モデル 単語に関する情報が格納されている．Sense モデルと一対多の関係で，複数の Sense モデルへのリレーションを持っており，Sense モデルを経由して，Synset モデルへのリレーションを辿ることで，単語からその定義を取得する．場合に応じて適切な言語のレコードを活用するため，言語での絞り込みをするスコープを持つ．

Synset モデル 単語の定義に関する情報が格納されている．主キーは文字列型の `synset` カラムである．Sense モデルと一対多の関係で，複数の Word モデルへのリレーションを持っており，Sense モデルを経由して，Word モデルへのリレー

ションを辿ることで、特定の定義をもつ単語を取得する．Word モデルと同様に、言語での絞り込みをするスコープを持つ．

Sense モデル Sense モデルは、上記の 2 つのモデルそれぞれと一対多のリレーションを持っている．synset カラムと wordid カラムが格納されており、それらを外部キーとして使用している．カラムと 2 つのモデルの双方向に紐づける役割を担う．単語と定義間で実行される処理は、Sense モデルを経由する．

4.2 提案する手法 ・ アプローチ

本研究での手法の流れを以下に示す．始めに、カテゴリ化したいドキュメントファイルをシステムにアップロードする．その後、ファイルに対して OCR 処理をして、内容の単語群を取得する．OCR と親和性のある代表的なプログラミング言語として Python が挙げられるが、http リクエストによって言語を問わず簡単に利用できる点や、OCR 処理に対するオプションの豊富さから、Google Vision API を本研究で使用する．その後、WornNet データベースと連携し、単語ごとの定義を取得する．WordNet は日本語、英語双方に対応した対規模な語彙データベースであるため、言語の入り混じったドキュメントに対しても問題なく処理をすることができるため、対応可能な語彙の量と汎用性を加味し、本研究で使用する．各々の定義を元に単語がどのカテゴリに属するかを判別し、単語ごとにラベリングをする．ラベリング結果から、最終的なドキュメントのカテゴリを決定する．なお、各々のカテゴリに関しては、WordNet の Synset モデルにカテゴリ名を入力し、単語群を取得する方法で用意している．

4.2.1 ドキュメントファイルをアップロード

Web 上で動く Ruby on Rails のシステムに対して、カテゴリ化したいドキュメントのファイルをアップロードする．ファイルのアップロード用のアップローダークラスを定義し、詳しいオプションの設定をする．アップロードされた画像ファイルのインスタンスをサービスクラスに渡し、OCR 等の処理をする．上記の処理の流れをソースコード 4.1 に示す．

```
1      <%= form_with(model: @document, url: '/', method: :post,
2                  multipart: true) do |form| %>
3          <div class="form">
4              <%= form.file_field :document_file %>
5          </div>
6          <%= form.submit 'アップロード' %>
7      <% end %>
```

ソースコード 4.1: フロントエンドの ERB

4.2.2 OCR によってドキュメントの内容を取得

Google Vision API の OCR 機能を用いて、ドキュメントの内容を取得し、単語ごとに分割する。本研究では、まず OCR を実行するために Google Vision API を活用し、外部サービスと連携するためのサービスクラスを設計する。このサービスクラス内で、API リクエストの送信処理やレスポンスの解析処理を行う。

まず VisionOcrService クラスのインスタンスを初期化する際に、画像認識のための ImageAnnotator::Client インスタンスを生成し、OCR を行う画像のパスを取得する。次に、指定された画像ファイルの内容をバイナリ形式で読み込み、Vision API にリクエストを送信する。API のレスポンスは、テキスト認識結果を保持するオブジェクトとして返される。このオブジェクト内から、OCR 結果として取得した文字列データをプロパティから抽出し、単語の配列として整形する。処理の最終段階では、API 通信時のエラー処理として、ネットワークエラーや API 側の制限に伴う例外処理を行い、適切なエラーメッセージをログに記録することで、システムの信頼性を向上させる。詳細な処理の流れをソースコード 4.2 に示す。

```
1   require "google/cloud/vision/v1"
2
3   class VisionOcrService
4     def initialize(image_path)
5       @image_path = image_path
6       @vision = Google::Cloud::Vision::V1::
7         ImageAnnotator::Client.new
8     end
9
10    def detect_text
11      image_content = File.binread(@image_path)
12      response = @vision.text_detection(image: {content:
13        image_content})
14
15      response.resources.flat_map do |res|
16        res.text_annotations.map(&:description)
17      end
18
19      rescue StandardError => e
20        Rails.logger.error "Vision API Error: #{e.message}"
21        []
22      end
23    end
```

ソースコード 4.2: Ruby による OCR の実装

4.2.3 旧字体を新字体に変換

分割した単語に旧字体が含まれていた場合、それらを新字体に変換する。この処理によって、OCRにより取得したドキュメントに含まれる旧字体が新字体へと効率的かつ高速に置換され、システム全体の信頼性や正確性の向上につながる。

変換前の旧字体文字列と変換後の新字体文字列を Font クラスに定数として定義し、変換対象となる文字とその一対一の対応関係を明確にする。なお、旧字体と新字体のデータは旧字新字対照表を参照している [15]。Analyzers::OldFontAnalyzer モジュールの convert メソッドでは、受け取ったテキストに対して Ruby の String#tr メソッドを用い、Font クラスで定義された旧字体と新字体の文字列を基に一括変換を実施する。詳細な実装例はソースコード 4.3 および 4.4 に示す。

例えば、経理カテゴリの「會計」という旧字体を「会計」という新字体に変換することで、古いドキュメントや、昔の文書を参照しているドキュメントに対しても適切なラベリングが可能になり、より汎用的なシステムになる。

```
1  class Font
2      BEFORE_CONVERT = "亞惡壓圍醫爲壹逸隱榮營衛驛謁圓鹽緣艷..."
3      AFTER_CONVERT  = "亜悪圧囲医為壹逸隠栄営衛駅謁円塩縁艶..."
4  end
```

ソースコード 4.3: Font クラスの定義

```
1  module Analyzers::OldFontAnalyzer
2      def convert(text)
3          before_convert_text = Font::BEFORE_CONVERT
4          after_convert_text  = Font::AFTER_CONVERT
5
6          text.tr(before_convert_text, after_convert_text)
7      end
8  end
```

ソースコード 4.4: 新字体への変換

4.2.4 それぞれの単語の定義の取得

変換後の単語に対し、WordNet のデータベースを用いて定義を取得する。まず OCR の結果を配列形式でインスタンス変数に格納し、その後、分析プロセスを開始する。

具体的には、まず単語リストに対して、WordNet の Word テーブルから該当するエントリを検索し、関連する定義情報を取得する。定義情報が見つからない場合は、"カテゴリ無し" として処理する。これらの処理は、OCR 結果のすべての単語に対して適用される。最終的に、Synset テーブルの一覧をもとに、文書のカテゴリを判定するためのデータが整えられる。詳細な処理の流れをソースコード 4.5 に示す。

```
1  class AnalyzeController < ApplicationController
2    before_action :set_words, only: [:analyze]
3
4    def analyze
5      @category_labels = Hash.new(0)
6
7      @words.each do |word|
8        analyze_word = Word.includes(:synsets).find_by(
9          lemma:
10         word)
11      return showNoCategoryError if analyze_word.nil?
12
13      result_words = analyze_word.synsets.pluck(:name)
14      current_labels = label_category(result_words)
15
16      current_labels.each do |category, count|
17        @category_labels[category] += count
18      end
19    end
20    @result_category = get_category(@category_labels)
21  end
22
23  private
24  def set_words
25    @words = @ocr_response
26  end
end
```

ソースコード 4.5: ActiveRecord による WordNet との連携

4.2.5 単語ごとにカテゴライズ，ラベリング

取得した単語の定義を基に，文書を適切なカテゴリに分類するためのラベリングを行う．カテゴライズ処理では，WordNet から取得した Synset 情報をもとに，どの単語がどのカテゴリに属するかを決定する．システムは，各 Synset を解析し，それがどのカテゴリに該当するかを判別する．この過程では，単語リストとカテゴリのマッピングをハッシュ形式で保持し，各カテゴリのカウントを増加させることで，どのカテゴリが最も関連性が高いかを測定する．

具体的には，単語を一意の集合として扱い，カテゴリごとの Synset 情報を取得し，対応するカテゴリラベルの数をカウントする．これにより，文書内の単語がどのカテゴリに属しているかを判別し，最終的な文書のカテゴリ決定へと繋げる．詳細な処理の流れをソースコード 4.6 に示す．

```
1  def label_category(words)
2    words_set = words.to_set
3
4    categories = Category.all
5    words_with_synsets = Word.where(lemma:
6      categories.pluck(:value)).includes(:synsets)
7
8    synsets_by_category = words_with_synsets
9      .each_with_object({}) do |word, hash|
10      hash[word.lemma] = word.synsets.map(&:name)
11    end
12
13    categories.each_with_object({}) do |category,
14      category_labels|
15      synset_names = synsets_by_category[category.value] ||
16        []
17      label_count = synset_names.count { |name| words_set
18        .include?(name) }
19      category_labels[category.value] = label_count
20    end
21  end
```

ソースコード 4.6: カテゴリのラベリングメソッド

4.2.6 ラベリング結果から文章のカテゴリを決定

ラベリングされた結果をもとに文書の最終的なカテゴリを決定する。すべての単語に対してカテゴリが割り当てられた後、カテゴリごとのラベルカウントを集計し、最も多く出現したカテゴリを文書の主要カテゴリとして選定する。

まず受け取ったラベルデータのハッシュを解析し、カウント数が最大のカテゴリを特定する。この処理の際、カテゴリが全く見つからなかった場合には、エラーをスローし、適切なメッセージを表示することで、未分類の文書に対する処理を明確化する。最終的に、識別されたカテゴリは、文書全体のカテゴリとして出力される。詳細な処理の流れをソースコード 4.7 に示す。

```
1   def get_category(labels)
2     max_label = labels.max_by { |_, value| value }
3     if max_label[1] == 0
4       showNoCategoryError
5     else
6       return max_label[0]
7     end
8   end
9
10  def showNoCategoryError
11    return 'no category'
12  end
```

ソースコード 4.7: 文書のカテゴリを決定するメソッド

第5章 実験・評価

実際の業務環境における多様な文書形式に対する本システムの有効性を評価するため、実験用のカテゴリとドキュメントを用意し、カテゴリライズを実施した。結果から、タイトル、カテゴリごとにドキュメントが適切なカテゴリに分類されているかを評価した。

5.1 実験で使ったカテゴリ・ドキュメント

本研究では、経理、人事、庶務、営業、エンジニアリングの5カテゴリを対象として実験を実施した。これらのカテゴリは、組織運営における主要な業務部門を網羅しており、各カテゴリごとにそれぞれ固有の専門用語やドキュメントの構造・フォーマットが異なる。例えば経理は会計関連の数値や伝票、人事は採用や労務管理に関する記述や採用の書類、庶務は各種連絡事項や管理文書、営業は提案書や顧客情報、エンジニアリングは技術仕様や設計書などが主となる。そのため、分類アルゴリズムの精度および汎用性を包括的に評価する上で十分なサンプル群であるといえる。

また、ドキュメントに関しては、各カテゴリに対してドキュメントを10個ずつ用意した。具体的には、企業で実際に利用される形式に類似したドキュメントを用いることで、システムの実務適用性を担保した。また、フォントや字体、縦書き・横書きといった文書構成のバリエーションが多くなるように選定し、分類アルゴリズムがこれらの要素に対して正常に動作するかどうかの評価を実施した。

5.2 精度の分析

5.2.1 タイトルごとの精度

タイトルごとの実験結果を表 5.1 ～ 5.5 に示す。

表 5.1: 経理カテゴリの精度

| タイトル | 結果 | 備考 |
|-------------------|----|----------------|
| 会社経理統制と経理検査 | 成功 | 昔の字体が存在する |
| 工業経理規範 | 成功 | 字体は全て昔の書き方 |
| 戦時会社経理統制体制の展開 | 成功 | 縦書き 2 列構成 |
| 法人税法の損金経理要件について | 成功 | 減価償却等の専門的な情報 |
| 陸軍経理組織の変遷と内部監査制度 | 成功 | |
| 企業における ERP システム導入 | 成功 | |
| アスピレーションの欠如と管理会計 | 成功 | |
| 中小企業の内部統制の進め方 | 失敗 | 内部統制に関する専門的な情報 |
| 日本の中小企業の会計情報システム | 失敗 | 情報システム関連の単語が混在 |
| 中小企業の経営情報収集 | 成功 | |

表 5.2: 人事カテゴリの精度

| タイトル | 結果 | 備考 |
|--------------------|----|-----------------|
| トヨタウェイと人事管理・労使管理 | 成功 | 縦書き 2 列構成 |
| 公務員の人事異動と人材形成 | 成功 | 図や英語が混在 |
| 新・人事労務管理 | 成功 | フォントがゴシック体の太文字 |
| 人事労務管理 | 成功 | 縦書きと横書きが混在 |
| 戦略的パートナーとしての日本の人事部 | 成功 | 図や見出しが英語表記 |
| 人事制度に関する総合調査 | 成功 | |
| 中小企業の従業員満足度 | 成功 | |
| 中小企業の新卒採用行動 | 成功 | |
| 日本企業におけるタレントマネジメント | 成功 | |
| 中小企業の人材育成 | 失敗 | 図やグラフを使用した細かい分析 |

表 5.3: 庶務カテゴリの精度

| タイトル | 結果 | 備考 |
|-----------------------|----|-----------------|
| RPA が事務職に及ぼす影響に関する一考察 | 成功 | |
| 一般事務女性の職業生活意識に関する一考察 | 成功 | 見出しのみ英語表記 |
| 女性事務職に得る派遣労働者の活用 | 成功 | 縦書き 2 列構成のゴシック体 |
| 女性事務職のキャリア拡大と職場組織 | 成功 | |
| 女性事務職の賃金と就業行動 | 成功 | |
| 事務作業における RPA の進展 | 失敗 | エンジニアリング領域の単語 |
| 事務職のストレス状況調査 | 成功 | |
| 女性事務職のキャリア形成 | 成功 | |
| ミクロデータを用いた女性事務職 | 成功 | |
| 大学の事務業務と効率化 | 失敗 | 情報システム関連の単語 |

表 5.4: 営業カテゴリの精度

| タイトル | 結果 | 備考 |
|-------------------|----|-----------------|
| CRM 主要成功要因 | 成功 | CRM 関連の専門的な情報 |
| チームワーク型営業のパフォーマンス | 成功 | |
| デジタル化と顧客経験価値 | 失敗 | DX 分野の内容が多い |
| 営業組織のスキル可視化 | 成功 | 新人研修で使用するスライド形式 |
| 改善思考の営業プロセス管理 | 成功 | 縦書き 2 列構成 |
| 企業における営業力と管理体制 | 成功 | |
| 企業の営業力向上のために | 成功 | |
| 中小企業の営業組織強化法 | 成功 | |
| 小企業実態基本調査 | 成功 | |
| 日本における「営業」と「販売」 | 成功 | 縦書き 2 列構成 |

表 5.5: エンジニアリングカテゴリの精度

| タイトル | 結果 | 備考 |
|---------------------|----|-----------|
| アジャイル開発の成功度 | 成功 | 英語と日本語が混在 |
| 組織的なソフトウェアプロセス改善 | 成功 | |
| 大規模組織のソフトウェアプロセス改善 | 成功 | 縦書き 2 列構成 |
| 中小企業のソフトウェア品質改善 | 成功 | スライド形式 |
| 中小企業の IT 活動 | 成功 | |
| 中小企業のアジャイル開発のすすめ | 成功 | |
| プロジェクトに対するリスク分析 | 成功 | |
| 円滑なマネジメントの失敗要因 | 成功 | 縦書き 2 列構成 |
| プロジェクトの失敗回避に向けた分析 | 成功 | |
| 日本企業のオープンソース・ソフトウェア | 成功 | 英語と日本語が混在 |

表 5.1 ～ 5.5 に示したように、処理精度にはタイトルごとにばらつきが見られる。特に、内容の専門性が高い文書や、他の分野の情報を多く含んでいる文書では処理が失敗する傾向が確認された。一方で、縦書きや横書きが混在する文書、英語表記が一部含まれる文書、ゴシック体や太文字を用いた文書などにおいては、ほとんど問題なく処理が成功している。これは、現代の標準的なフォーマットに近い文書に対しては、OCR の処理が適切に機能することを示している。また、縦書き 2 列構成や図表が含まれる場合も成功率は高く、特に人事、営業、エンジニアリング関連のタイトルにおいては安定した処理精度が見られた。これらの結果から、フォーマットや字体の違いが OCR の精度に与える影響が明確になった。

5.2.2 カテゴリごとの精度

続いて、文書をカテゴリに分類し、それぞれのカテゴリごとに処理精度を評価した結果を表 5.6 に示す。

| カテゴリ | 精度 (%) |
|----------|--------|
| 経理 | 80 |
| 人事 | 90 |
| 庶務 | 80 |
| 営業 | 90 |
| エンジニアリング | 100 |

表 5.6: カテゴリごとの精度

表 5.6 に示したように、経理カテゴリにおける精度が他のカテゴリと比べて低いことが確認された。この主な要因としては、経理カテゴリの文書に特殊な専門用語が多く含まれている点が挙げられる。一方で、エンジニアリングカテゴリにおいては、フォーマットが比較的一般的であり、他のカテゴリの内容が含まれていることが少ないため、100%の精度を達成している。さらに、精度が高いカテゴリにおいても、処理内容には軽微な誤解釈が存在する場合があるが、全体としてカテゴリごとの分類精度に影響を与えるような問題は確認されなかった。これにより、OCR 処理のカテゴリごとの安定性について評価することができる。以上の結果から、タイトルごとの処理精度とカテゴリごとの精度において、それぞれの特性が確認された。本研究では、特定の条件下における処理の成功率を明らかにしたが、さらなる精度向上を目指すためには、特殊なフォーマットへの対応が課題として残る。これらの詳細な考察については次章で述べる。

第6章 考察

6.1 ドキュメントごとの結果の解釈

第5章で本研究で実施した実験を紹介したが、本章ではその結果から、本研究で提示したシステムの優位点及び将来性や有用性について述べる。まず、実験により明らかとなった本システムの優位点について述べる。それは、ドキュメントの形式に関わらず同様の処理結果を得ることができる点である。文書には、縦書きや横書き及びそれらの複合など、内容の形式はジャンルによって多岐にわたる。それら全てに対応出来なければ、本システムの優位性は著しく低下してしまう。しかし、さまざまな形式の文書を実験では用意したが、特に形式の差による精度の変化は見受けられなかった。そのため、本システムにおいては、文書をアップロードする前に形式を変換する必要がなく、その点ではストレスなくシステムを運用することができると考えられる。本システムの将来性については、発展途上であるという結論になってしまうが、将来的にカテゴリが増え、解析の幅が広がることや、WordNet データベースがアップデートされ、扱うことのできる語彙が増えていくことを考えると、IT 分野の発展に伴い、相対的にニーズが増えていくと考えられる。本システムの有用性については、第1章で大まかに述べたが、紙媒体中心の業務に不便を感じている人が現代に多くいるため、そのような状況を改善するという観点では、要件を満たすことができているといえる。

6.2 改善点

本システムの改善点について挙げられることは、WordNet データベースに存在しない専門的な単語を多く含む文書に関しては、精度が落ちてしまう点である。この欠点を改善するためには、現状使用している WordNet データベースの他に、各々のカテゴリに即した専門的なデータベースを使用し、判別可能な語彙を拡張することが必要である。内容の異なる複数のデータベースを1つの MVC アーキテクチャに統合することができれば、本システムの有用性や汎用性の向上につながると考えられる。また、現状のシステムでは、日本語と英語を主な対象としており、WordNet データベースを利用してカテゴリの解析をしている。しかし、グ

ローバル化が進む現代社会では、多言語対応が求められる場面が増えている。特に、アジア地域で広く使用されている中国語、韓国語に対応することで、システムの利用範囲をさらに拡大することが可能である。これを実現するためには、各言語に特化した語彙データベースを統合し、多言語間でのカテゴリ解析を可能にするアルゴリズムの開発が必要である。

第7章 おわりに

7.1 研究のまとめ

本研究で開発したシステムは、現在の業務フローにおける課題を解決し得る有効なツールであると結論付けられる。また、さらなる改良を通じて、専門分野や多言語対応、クラウドサービス化等の新たな領域への応用が期待される。現代では、今後も IT 分野の著しい成長に遅れをとった企業を中心に、さまざまな業務的ニーズが発生すると考えられる。本研究では紙媒体中心業務をさらに細分化した、ドキュメントのカテゴリ分け分野に着目してシステムの作成をしたが、本システムのようなソリューションが生み出されることで、IT 社会の課題が解決されることを願っている。

7.2 今後の展望

本システムを現実的な業務フローに組み込むには、ユーザーインターフェース (UI) の最適化も重要である。現状ではシステムのコア機能に重点を置いているが、操作の直感性やユーザーエクスペリエンスの向上が求められる。例えば、解析結果を可視化するダッシュボードや、フィードバック機能を備えたインターフェースを実装することで、利用者の利便性を高めることが可能である。処理の精度について、経理、人事、庶務等のカテゴリにおいて、高い精度での解析が可能であることが確認された。本システムはさらに分野ごとに特化した最適化を施すことで、専門分野への応用を推進できる、例えば医療分野や法律分野では特有の専門用語や文書構造が存在するため、分野別にカスタマイズされた解析アルゴリズムが必要である。また、本システムをクラウドベースで提供することにより、複数のユーザーが同時に利用できる環境を構築し、スケーラビリティを確保することができる。また、API を公開することで、他のアプリケーションやサービスとの統合を可能にし、システムの拡張性をさらに高めることも有効である。

謝辞

本研究の遂行にあたり，多大なるご指導とご助言を賜りました指導教員の Martin J. Dürst 教授に心より感謝申し上げます．研究の方向性について貴重なご意見をいただき，また研究内容の精査においても適切なアドバイスをいただきましたことに深く感謝いたします．また，研究活動において協力を惜しまなかった研究室の皆様へ感謝申し上げます．議論を通じて新たな視点を得ることができ，研究の深化に大いに寄与していただきました．さらに，本論文の作成にあたり添削をしてくださった石井幹大助手に心より感謝申し上げます．皆様の支えがなければ，本システムの開発は実現し得なかったことを強く認識しております．本研究が，多くの人々のご支援とご協力によって成り立っていることを改めて認識し，深く感謝の意を表します．

参考文献

- [1] 佐々木康浩. RPA (Robotic Process Automation) の可能性. 経営情報学会 全国研究発表大会要旨集, Vol. 2017s, pp. 201–204, 2017.
- [2] D-Analyzer. RPA にできないこと, 不得意な業務はなにか? https://tepss.com/danalyzer/column/rpa_can_not_do.html, 2019. 参照日 2024 年 12 月 12 日.
- [3] 伊藤司. 電子帳簿等保存制度の主な沿革と今後の展望. 立法と調査, No. 461, pp. 138–152, Nov 2023.
- [4] まつもと ゆきひろ. オブジェクト指向スクリプト言語 Ruby. <http://www.ruby-lang.org/ja/>, 2023. 参照日 2024 年 12 月 12 日.
- [5] Noel Rappin and Dave Thomas. *Programming Ruby 3.3: The Pragmatic Programmers' Guide (Pragmatic Programmers; Facets of Ruby)*. Pragmatic Bookshelf, 5th edition, 2024.
- [6] 高橋明日香. フレームワークを用いた Web アプリケーション開発における変更容易性の評価. 第 112 巻, pp. 115–120, 2013.
- [7] 大西淑雅. 文字認識 API を用いた講義アーカイブシステムの設計. 第 33 巻, pp. 25–30, 2018.
- [8] 竹内孔一. 日本語 Wordnet における語義・概念の分散表現獲得. 第 18 巻, pp. 245–246, 2019.
- [9] 倉橋和子. 分割・併合機能を有する k-means アルゴリズムによるクラスタリング. 第 106 巻, pp. 67–71, 2007.
- [10] 東埜淳哉, 幸裕. 階層的クラスタリングを用いたクラスタ数の自動推定に関する検討. 日本知能情報ファジィ学会 ファジィ システム シンポジウム 講演論文集, Vol. 38, pp. 679–684, 2022.
- [11] Somayya Madakam, Rajesh M. Holmukhe, and Durgesh Kumar Jaiswal. The future digital work force: Robotic process automation (RPA). *Journal of Information Systems and Technology Management*, Vol. 16, No. 1, Jan. 2019.

- [12] Lee Kasowaki and Eden Jack. Robots at work: Robotics process automation and the future of employment. EasyChair Preprint 11256, EasyChair, 2023.
- [13] 花房亮, 山岸秀一, 松本慎平, 加島智子. 機械学習処理に基づいたプログラミング読解中の視線軌道の自動分類. 人工知能学会全国大会論文集, Vol. JSAI2015, pp. 3N32–3N32, 2015.
- [14] 松田勝志, 福島俊一. 文書タイプ分類による問題解決向き www 検索システムの開発と評価. Technical Report 20(2007-FI-053), NEC ヒューマンメディア研究所, NEC ヒューマンメディア研究所, Mar 2007.
- [15] kameique.com. 旧字新字対照表. <http://www.kameique.com/tool/kanji34/kyujitable.html>. 参照日 2025 年 2 月 12 日.

付録

- A. 質疑応答
- B. 実験で使⽤したドキュメント
- C. カテゴリズの判別基準となる単語群

付録 A

質疑応答

大原先生の質問 (要約)

業務文書はタイトルでキーワードが選別でき、数サンプル与えれば機械学習でカテゴリ化出来てしまうが、あえて WordNet を使用する理由はなにか。

当日の回答

タイトルや内容が曖昧なドキュメントも、文章全体を見て正しくカテゴリ化するために、WordNet を使用している。また、コストの面で機械学習モデルを用いた関連研究との差別化を行うために、カテゴリ化のアルゴリズムを自ら考える必要があると判断した。

森田先生の質問 (要約)

1 つの単語に複数の定義が所属しているが、単語の多義性の部分で何か行っていることはあるか。

当日の回答

単語の定義に該当するものを全てカテゴリ化の判別基準として使用しており、カテゴリに紐づく定義を含む単語をキーワードとして使用している。

森田先生の質問 (要約)

文脈によって意味が変わる単語も存在するか、定義の取得はどのようにしているか。

当日の回答

今回単語はカテゴリ名に付随しており、キーワードという形でアルゴリズムに適用しているため、文脈によって変わるような難しい単語ではないと判断している。

シュデシナ先生の質問 (要約)

紙媒体が好きな人もいるが、このシステムを使うユーザーはどのような人か.

当日の回答

紙媒体を使用しているが、IT 知識の乏しさによってデジタル化したくてもできない企業が主なターゲットになっている.

付録B

実験で使ったドキュメント

経理

久保田秀樹, 「会社経理統制令と経理検査」, 山内隆教授退官記念論文集, No.329, pp.169-172, 1934 年

大野巖, 「工業経理規範」, 昭和 13 年 9 月, 化学機械, Vol.2 No.3, 1938 年

柴田雅, 「戦時会社経理統制体制の展開」, The Socio-Economic History Society, 1937 年

前原真一, 『法人税法の損金経理要件について』, 税務大学校, 研究部教授

建部宏明, 「陸軍経理組織の変遷と内部監査制度Ⅱ」, 明治大学経理研究所, 2009 年

土山真由美, 「企業における ERP システム導入の課題と取り組み」, 開発工学 Vol.41 No.2, 2021 年

黒木淳, 「経営者のアスピレーションの欠如と管理会計の実践度」, 会計プロGRESS No.21, 2020 年

渡邊公認会計士税理士事務所, 「中堅・中小企業のための実践的内部統制の進め方のポイント総点検」, 内部統制報告書, 2008 年

山田恵, 「わが国法人中小企業の会計情報システムに関する実証的研究」, 管理会計学 Vol.8 No.1-2, 2000 年

独立行政法人 中小企業基盤整備機構, 「中小企業経営者の経営情報の収集・活用に関する実態調査」, 中小機構調査研究報告書 Vol.5 No.7, 2013 年

人事

猿田正機, 「トヨタウェイと人事管理・労使関係」, 税務経理協会, 2007 年

圓生和之, 「公務員の人事異動と人材形成—大卒ホワイトカラーの公民比較からの分析」, 日本労働研究雑誌, No.759, pp.47-53, 2023 年

津田眞激, 「新・人事労務管理」, 有斐閣, 1995 年

津田眞激編著, 「人事労務管理」, ミネルヴァ書房, 1993 年

平野光俊, 「戦略的パートナーとしての日本の人事部—その役割の本質と課題—」, 神戸大学大学院経営学研究科, 2010 年

産労総合研究所, 「第 8 回 人事制度等に関する総合調査」, 2021 年

関田良彦, 「中小企業における従業員の満足度とモチベーションに関する調査分析」, 高知工科大学マネジメント学部研究報告, 2017 年

山本和史, 「中小企業における新卒採用行動に関する実証分析」, 日本労務学会誌 Vol.18 No.1, 2017 年

佐藤俊一, 「中小企業における人材育成（企業内研修の事例研究）」, 日本経営診断学会第 53 回全国大会, 2020 年

柿沼英樹, 「日本企業におけるタレントマネジメントの展開と現状」, Works Discussion Paper No.4, 2015 年

庶務

所正文, 「一般事務職女性の職業生活意識に関する一考察」, 経営行動科学, Vol.3 No.1, 1988 年

古武真美, 「女性事務職における派遣労働者の活用」, 近畿大学短期大学論集, Vol.44 No.1, pp.11-20, 2011 年

浅海典子, 「女性事務職のキャリア拡大と職場組織」, 日本経済評論社, 2006 年

寺村絵里子, 「女性事務職の賃金と就業行動」, 国際短期大学人口学研究, No.48, 2012 年

日高義浩, 「RPA が事務職に及ぼす影響に関する一考察」, 鹿児島経済論集, Vol.63 No.1, 2022 年

岩本隆志, 「事務作業における RPA の進展について」, 日本生産管理学会論文誌 Vol.27 No.1, 2020 年

洲崎好香, 有吉浩美, 山田英津子, 熊井三治, 「事務職者におけるストレス状況調査」, 日本健康医学協会誌, Vol.17 No.1, 2008 年

駒川智子, 「女性事務職のキャリア形成と『女性活用』」, 大原社会問題研究所雑誌 No.582, 2007 年

寺村絵里子, 「女性事務職の賃金と就業行動」, 統計数理研究所, 2011 年

両角亜希子, 王帥, 「大学の事務業務とその効率化の規定要因」, 広島大学 高等教育研究開発センター, 大学論集第 55 集, 2023 年

営業

高彰培,「CRM 主要成功要因と成果間の関連性に対する実証的研究」, NAIS Journal, 2022 年

松尾睦, 早川勝夫, 高嶋克義,「改善志向の営業プロセス管理 ― 日本ベーリンガーインゲルハイムの事例」, マーケティングジャーナル Vol.30 No.3, 2011 年

小須田庸平,「企業の営業力向上についての考察 ― インターナルマーケティングアプローチの視点から」, 企業研究報告, 2022 年

橋本佳子,「中小企業経営改善のための営業組織強化手法」, 中小企業診断報告書, 2006 年

中小企業庁,「中小企業実態基本調査」, 令和 5 年度, 2023 年

山崎淳一, 高野研一,「チームワーク型営業のパフォーマンス要因」, 慶應義塾大学大学院 システムデザイン・マネジメント研究科, 2021 年

張淑梅,「デジタル化がもたらす顧客経験価値の進化に関する一考察」, 日本福祉大学経済論集, No.67, 2023 年

柴山雄太,「営業組織・営業担当者の能力開発におけるスキルの可視化」, 株式会社ライトワークス, 2024 年

清水良郎,「企業における営業力とその管理体制についての考察」, 名古屋学院大学論集 社会科学篇, Vol.52 No.4, 2016 年

山内孝幸,「日本における『営業』と『販売』に関する考察」, 阪南論集 社会科学編, Vol.56 No.1, 2020 年

エンジニアリング

今仁武臣, 中野冠, 「アジャイル型開発手法の適用領域とプロジェクトの成功度の関係」, 日本情報経営学会誌 Vol.37 No.150, 2017 年

小室睦, 鷺崎弘宣, 「ソフトウェアプロセス改善を組織的、実証的にすすめるためのデータ分析パターン言語の提案」, ソフトウェア品質シンポジウム 2024, 2024 年

小笠原秀人, 藤巻昇, 艸薙匠, 田原康之, 大須賀昭彦, 「大規模組織におけるソフトウェアプロセス改善活動の適用評価—10 年間の実践に基づく考察」, 情報処理学会論文誌 Vol.51 No.9, 2010 年

濱崎利之, 水本継, 横山晃生, 「地方中小企業におけるソフトウェア品質改善の取り組み」, NDKCOM, 2024 年

井上博進, 金沢健, 「中小企業における IT 活用の実態と課題」, 愛知工業大学 研究報告, 1999 年

田中宏和, 「中小企業を対象にしたアジャイル開発の進め方」, 静岡大学, 2015 年

保田洋, 「中小企業のプロジェクトマネジメントに対する課題解決のためのリスク分析」, 甲子園短期大学紀要 37, 2019 年

保田洋, 川向肇, 西村治彦, 「中小企業プロジェクトの円滑なマネジメントに向けた失敗要因の分析」, 情報知識学会誌 Vol.30 No.3, 2020 年

保田洋, 川向肇, 西村治彦, 「中小企業プロジェクトの失敗回避に向けたリスク分析」, 第 28 回年次大会予稿, 2020 年

野田哲夫, 丹生晃隆, 「日本の IT 企業におけるオープンソース・ソフトウェアの活用・開発貢献が企業成長に与える影響に関する研究」, 経済科学論集, No.42, 2016 年

付録C

カテゴライズの判別基準となる単語群

経理

accounting

主計

会計

経理

会計学

簿記

アカウンティング

計数

accountancy

会計業務

人事

human resources

人事

採用

雇用

配置轉換

給与

労務管理

研修

人事考課

組織開発

福利厚生

人材育成

離職

人事評価

庶務

general affairs

庶務

総務

文書管理

資産管理

契約管理

社内便

来客対応

ファシリティ

備品管理

事務補助

受付業務

オフィスマネジメント

営業

sales

セールス

販売

顧客管理

販売促進

クロージング

提案営業

営業活動

商談

新規開拓

アポイントメント

営業目標

受注

営業戦略

エンジニアリング

engineer

software development

システム開発

プログラミング

ソフトウェア

コーディング

開発環境

テストエンジニア

デバッグ

ハードウェア

ネットワーク

技術支援

アルゴリズム

インフラストラクチャー

アーキテクチャ