

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

SNS からの意図しない個人情報漏えい検出 システムの開発

—アナライザ用周辺機能の開発及び GUI の設計—

孫 尚君

(コンピュータサイエンス専攻)

指導教員 田中二郎

2012年 3月

概要

本報告書では、学生提案による「SNS からの意図しない個人情報漏えい検出システムの開発」というテーマで、システムの企画、開発及び評価を学生 4 名で実施した結果及び筆者が担当した部分を詳しく述べる。

我々が「SNS からの意図しない個人情報漏えい検出システムの開発」というテーマを取り上げた理由は、近年、SNS の拡大とともに SNS 上のプライバシー問題が大きな問題になっていると感じ、プライバシー問題に対する SNS の利用者に注意の喚起をしたいと思っているからである。具体的には、近年 Twitter や Facebook などのソーシャル・ネットワーキング・サービス (SNS) を利用するユーザが増えている。その中で、利用者によっては特定の単語を使ったコメント等が多くなり、そこから本人が意図していない個人情報および属性などが不特定多数の人に知られてしまう危険性がある。SNS ユーザの発信したコメント等から、他人に自分の情報が特定され、万が一悪用されることを防止するという観点でシステム開発を行った。このシステムを利用することにより、SNS ユーザは、どのような属性や個人情報が特定されてしまうかを把握することができる。

開発において筆者は、アナライザ用周辺機能である言語判定機能と動的にクラスローディング機能の開発、システム GUI の開発、および性別推定アナライザの開発を担当した。言語判定機能は解析対象となるユーザの SNS アカウントから取得したつぶやきや発言等のデータがどんな言語で記述されているかを判断する機能である。言語判定機能の開発では、複数の言語判定ライブラリを組み合わせることで、単独ライブラリの言語判定より正解率が大幅に上昇した。動的にクラスローディング機能は、Java や Perl などの Python 以外の言語で実装された解析アルゴリズム (アナライザ) を呼び出すことを可能とするものである。システム GUI を開発する際には、ウェブページの見やすさ・使いやすさを考慮し、ユーザ認証ページと解析結果表示ページを作成した。性別推定アナライザは、日本語の特徴を通じて、ユーザの SNS での発言からユーザの性別を推定するものである。性別推定アナライザの開発では、筆者が提案した性別推定手法の有効性を検証するために、SNS 上での話し言葉の男女差の実験を行った。実験の結果を利用して、性別推定用のアルゴリズムを作製した。

目次

第 1 章	はじめに	1
1.1	本報告書の内容	1
1.2	システム開発の背景	1
1.2.1	SNS とは	1
1.2.2	SNS のプライバシー問題	1
1.2.3	SNS プライバシーに関する課題	1
1.3	プロジェクトの目的	2
1.4	本報告書の構成	2
第 2 章	個人情報漏えい検出システムの概要	3
2.1	システム要件	3
2.2	想定するユーザ	4
2.3	システムの構成	4
2.3.1	ハードウェア構成	4
2.3.2	ソフトウェア構成	4
2.3.3	開発環境	4
2.4	開発体制	5
第 3 章	個人情報漏えい検出システムの設計	6
3.1	設計方針・アーキテクチャ構成	6
3.2	各レイヤの機能	7
3.2.1	クローラレイヤの機能	7
3.2.2	アナライザレイヤの機能	8
3.2.3	ビューレイヤの機能	8
3.2.1	スケジューラの機能	9
3.3	システム動作	9
3.4	設計の詳細	10
3.4.1	クローラレイヤ	10
3.4.2	アナライザレイヤ	11
3.4.3	ビューレイヤ	11
3.4.4	スケジューラ	12
第 4 章	アナライザ用周辺機能及び GUI の設計	13
4.1	開発作業の分担	13
4.2	言語判定機能	14
4.2.1	言語判定機能の概要	14
4.2.2	既存ライブラリやサービスについて	15
4.2.3	言語判定機能の設計詳細	19
4.2.4	実験	20
4.3	動的にクラスローディング機能	24
4.3.1	動的にクラスローディング機能の概要	24
4.3.2	アナライザの実装	24

4.3.3	Java と Perl 言語への呼び出し	26
4.4	GUI の開発	27
4.4.1	画面の設計	27
4.4.2	画面の実装	27
4.4.3	ユーザ認証ページ	27
4.4.4	解析結果表示ページ	28
4.4.5	画面のテスト	30
第 5 章	性別推定アナライザの開発	31
5.1	アナライザの開発について	31
5.2	性別推定アナライザの概要	31
5.3	関連研究	32
5.4	性別推定の手法	32
5.5	実験	36
5.6	今後の課題	36
第 6 章	プロジェクトの経過と進め方	37
6.1	プロジェクトの進め方	37
6.2	開発スケジュール	37
6.2.1	開発スケジュールの計画	37
6.2.2	開発スケジュールの実績	38
第 7 章	システムの評価	39
7.1	開発メンバによるシステムの評価	39
7.2	第三者によるシステムの評価	40
7.2.1	評価の目的	40
7.2.2	評価の方法	40
7.2.3	評価の結果	40
第 8 章	おわりに	41
謝辞		42
参考文献		43
付録		45

図目次

図 3-1 システム設計方針（3 層設計）	6
図 3-2 システム設計方針（スケジューラとインタフェースと言語判定機能）	6
図 3-3 システムアーキテクチャ	7
図 3-4 一つのアナライザが一つの SNS データを解析に用いる	8
図 3-5 一つのアナライザが複数の SNS データを解析に用いる	8
図 3-6 Queue の使用	10
図 3-7 Django により結果表示	12
図 4-1 SNS での言語使用の例	14
図 4-2 言語判定機能利用の流れ	15
図 4-3 既存ライブラリとサービスの判定の正解率	17
図 4-4 既存ライブラリとサービスの判定の誤判定率	18
図 4-5 既存ライブラリとサービスが判定にかかった時間	19
図 4-6 単言語を含むつぶやきに対しての言語判定の流れ	21
図 4-7 類似度比較範囲の算出	21
図 4-8 多言語を含むつぶやきに対しての言語判定の流れ	22
図 4-9 言語判定実験の結果	23
図 4-10 manifest.xml ファイルの例	25
図 4-11 AbstractJavaAnalyzer クラスのサンプルコード	25
図 4-12 Python 言語から Java 言語のアナライザを呼び出す	26
図 4-13 画面配色の比率	27
図 4-14 ユーザ認証ページの画像	28
図 4-15 解析結果ページの画面（基本情報）	29
図 4-16 解析結果表示部分	30
図 5-1 性別推定の手法	33
図 5-2 確信度計算のアルゴリズム	33
図 5-3 性別推定の実験の結果	35
図 5-4 性別推定アナライザの正解率実験の結果	36
図 6-1 開発スケジュールの計画	37
図 6-2 開発スケジュールの実績	38

表目次

表 2-1 システムのハードウェア構成.....	4
表 2-2 システムのソフトウェア構成.....	4
表 2-3 開発環境	5
表 2-4 プロジェクトの体制	5
表 3-1 SNS ユーザに通知する個人情報.....	9
表 4-1 開発作業の分担	13
表 4-2 既存の言語判定ライブラリ・サービス	15
表 4-3 画面テストで使うブラウザ	30
表 4-4 画面テストで使う解像度(ピクセル)	30
表 5-1 アナライザ開発担当の分担	31
表 5-2 SNS での代名詞と終助詞の出現頻度	34
表 5-3 性別推定の確信値 (抜粋)	35
表 7-1 アンケートの質問項目の一部.....	40

第1章 はじめに

1.1 本報告書の内容

本報告書は筑波大学システム情報工学研究科コンピュータサイエンス専攻「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」における特定課題研究として、学生が提案した「SNS からの意図しない個人情報漏えい検出システム」というテーマで、システムの企画、開発及び評価を学生 4 名で実施した成果をまとめたものである。

本報告書では、システム開発の概要、実現手法及び成果について報告するとともに、特にシステム開発段階で筆者が担当したアナライザ用周辺機能である言語判定機能と動的にクラスローディング機能の開発、システム GUI の開発、および性別推定アナライザの開発について詳しく報告するものである。

1.2 システム開発の背景

1.2.1 SNS とは

ソーシャル・ネットワーキング・サービス（英語：Social Network Service、SNS）とは、人と人とのつながりを促進・サポートする Web サービスのことである。代表的な SNS として、日本最大の 2535 万（2011 年 9 月）の会員数を持つ mixi、モバイル向けの GREE、モバゲータウン、海外では世界最大の 8 億人（2011 年 9 月）の会員を持つ Facebook、それに次ぐ Twitter、Myspace などがある。

1.2.2 SNS のプライバシー問題

近年、SNS の拡大とともに増すプライバシー問題に対する意識が高まっている[1]。たとえば、実名登録が必須の Facebook はプライバシー設定機能をユーザに提供しているが、プライバシー設定が複雑で適切な設定が困難であることから、デフォルトの設定を利用するユーザも多く、ユーザが個人情報の公開範囲を把握できていないことがある。あるいは、匿名で登録している Twitter アカウントは、発言や友人関係を分析することで、ユーザの身元が推測されてしまう可能性もある。最近、日本では、次のような事例があった[2]。

アディダスジャパンに勤務する社員が、顧客の来店情報を Twitter に書き込み、誹謗中傷を行った。その情報がインターネット上に回わり、社員の過去のツイートを読み、社員の個人情報を特定、その社員は免職になった。

このような事例から、身元情報を直接公開していなくとも、これまでの SNS 利用状況などを解析することで個人に身元が推測されることがわかる（これを人物属性推定[3][4][5]という）。この事例は、推測された情報をもとに実際に個人が特定されうるといふ脅威の実例といえる（これを脱匿名化の問題という）。

1.2.3 SNS プライバシーに関する課題

我々が考えた SNS プライバシーに関する課題は以下の 2 点である。

1. SNS 上であるユーザの脱匿名化を行う攻撃手法の研究が知られている[6]。しかし、それらの研究成果をサービスとして SNS ユーザに提供している例は少ない。

2. 脱匿名化アルゴリズムの解析の精度や速度を向上させるための研究が今後活発になると予測されるが、SNS 情報の収集や、データの解析、解析結果の通知などの機能を備えたプラットフォームが研究者に望まれている。

1.3 プロジェクトの目的

本プロジェクトの目的は 1.2.3 節で記述する二つの課題を解決することである。具体的には、以下の二つの目標を実現する。

1. SNS ユーザに対して、ユーザの SNS 利用状況を解析し、どのレベルまで個人情報が抽出される可能性があるかをユーザに通知する Web サービスを作成する。これによりユーザが個人情報の漏えい状況を把握することができ、何かの対策を実施することを可能とする。
2. SNS のプライバシー問題の解決を目指す研究の手助けとなるプラットフォームを作成する。研究者がこのプラットフォームを運用することで、SNS ユーザの情報収集、推測アルゴリズムの実験などの仕事が容易にできるようになるも目標とする。

1.4 本報告書の構成

本報告書は全 8 章から構成されている。第 2 章では、本システムの概要として、システムの要件や想定する利用者、及びシステムの構成について述べる。第 3 章では、システムの設計方針やアーキテクチャ構成や設計の詳細について述べる。第 4 章では、筆者が担当したアナライザ周辺機能及びシステム GUI の開発について詳しく説明する。第 5 章では、筆者が開発した性別推定アナライザについて詳しく説明する。第 6 章では、プロジェクトの経過とスケジュールについて述べる。第 7 章では、開発したシステムの評価について述べる。第 8 章では、今回のシステム開発を通じて得られた感想や知識についてまとめる。

第2章 個人情報漏えい検出システムの概要

本章では、個人情報漏えい検出システムの概要について説明する。2.1 節では、システムの要件について述べる。2.2 では、システムの想定するユーザについて述べる。2.3 節では、システムのハードウェア構成とソフトウェア構成について詳しく述べる。2.4 節では、今回システム開発の体制について述べる。

2.1 システム要件

1.3 節で述べたプロジェクトの目的である「SNS 上での個人情報漏えい状況をユーザに通知する」と「SNS プライバシーに関する研究者の手助けとなるプラットフォームを作成する」を実現するために、本システムでは以下の 7 つの要件を満たすプラットフォームを作成する。

1. 解析結果をユーザに通知すること。

ユーザのアカウント情報と解析結果を Web ページで表示できる。また、ユーザ個人のアカウント情報と解析結果のみ通知できること。

2. 複数の SNS から情報収集を行えること。

本システムでは、ユーザの情報収集の対象となる SNS を、登録ユーザ数・知名度などを考慮し、Facebook と Twitter とすること。さらに、他の SNS や今後登場する新たな SNS からユーザ情報を収集する機能の追加が可能であること。

3. 複数の SNS を解析対象とすること。

1 の要件で取得した SNS 毎の情報の解析、あるいは複数の SNS からの情報を解析可能とすること。

4. 2 と 3 に関して追加が容易であること。

今後登場する新たな SNS に対するデータの収集・情報解析を行えること。また、アナライザの実装は Python・Perl・Java など複数の言語で行えること。

5. アナライザの組み合わせが容易に行えること

複数のアナライザの解析結果を組み合わせ、さらに解析を行うことができる。

6. SNS 上での使用言語を判定できること。

ユーザ情報の解析を行うために、言語依存のアナライザを選ぶ時、SNS 上でユーザが利用している言語を判定できることである。

7. 情報収集、解析のタイミングを制御できること。

初めてサービスに登録を行ったユーザに対して情報収集のクロール操作や情報解析を行うタイミングを制御できることである。

2.2 想定するユーザ

本システムで想定されるユーザを、以下に示す。

- SNS のユーザ：

Twitter や Facebook 上の個人情報の漏えい状況に関心を持っている者。

- 研究者：

SNS プライバシー問題に関する課題を研究していて、本システムを研究のツールとして利用したい人。

2.3 システムの構成

2.3.1 ハードウェア構成

本システムのハードウェア構成は、表 2-1 に示す。

表 2-1 システムのハードウェア構成

項目	名称
CPU	Intel 2.33GHz
メモリ	2.0GB
ハードディスク	250GB

2.3.2 ソフトウェア構成

システム開発の費用を減らすために、システムを構成するソフトウェアは、基本的にオープンソースソフトウェア（OSS）を利用とする。具体的には、表 2-2 の構成とする。Web システムの構成として Linux+Web サーバ+データベース+開発言語の構成である。ソフトウェアのバージョンに関しては、品質やセキュリティ等を考えてなるべく新しいバージョンを用いる。

表 2-2 システムのソフトウェア構成

項目	名称
OS	CentOS
Web サーバソフトウェア	Apache
データベースソフトウェア	MongoDB2.0
開発言語	Python2.7
フレームワーク	Django

2.3.3 開発環境

今回の開発環境は表 2-3 の通りである。

表 2-3 開発環境

項目	名称	バージョン
OS	CentOS	5.6 x86_64
開発言語	Python	2.6 / 2.7
	XHTML	1.0
	CSS	2.1
開発フレームワーク	Django	1.3
統合開発環境	Eclipse	3.5

本システムは Python 言語を用いて開発を行った。これは、Python 言語がシステム管理からデスクトップやウェブアプリケーションまで、幅広く応用可能なプログラミング言語であるためである。Python を使うことで、システムの画面からデータ処理まで、すべての機能を 1 つの言語で実現できる。

2.4 開発体制

プロジェクトは筆者を含めた高度 IT コースの学生 4 名がメンバーの中心として進めた。開発メンバーの関係とそれぞれの役割分担を表 2-4 に示す。また、課題担当教員である筑波大学コンピュータサイエンス専攻の山本幹雄教授・佐久間淳准教授が研究・開発の指導を行った。

表 2-4 プロジェクトの体制

役割	責任者	
	主担当	副担当
プロジェクトリーダー	田中	João Orui
技術リーダー	João Orui	長野
アーキテクチャデザイナー	長野	孫
GUI デザイナー	孫	田中
ドキュメント管理	長野	—

筆者は GUI デザイナーの主担当として、GUI デザインの作業を行った。また、デザイン作業が終了後、仕事量が多いメンバーに対して作業の支援を行った。具体的には、言語判定や動的にクラスローディングなどアナライザ周辺機能と性別判定アナライザの設計・実装を行った。

第3章 個人情報漏えい検出システムの設計

本章では、個人情報漏えい検出システムの設計について述べる。3.1 節では、システム全体の設計方針・アーキテクチャ構成について述べる。3.2 節では、システムの各レイヤの機能について述べる。3.3 節では、システム動作の流れについて述べる。3.4 節では、筆者担当部分の説明を行う際の前提知識として、各レイヤの設計詳細を述べる。

3.1 設計方針・アーキテクチャ構成

本システムは、2.1 節で述べた 7 つの要件を満たすことができるようにシステムを設計した。まず、要件 1 と要件 2 と要件 3 を満たすために、図 3.1 のように SNS 情報の収集を行うクローラレイヤ、情報解析を行うアナライザレイヤ、解析結果を表示するビューレイヤの 3 階層の独立した層からなるプラットフォームのアーキテクチャ設計を行った。また、要件 7 を満たすために、クローラレイヤ、アナライザレイヤ及びビューレイヤの動作タイミングの制御を行うためのスケジューラを設計した。

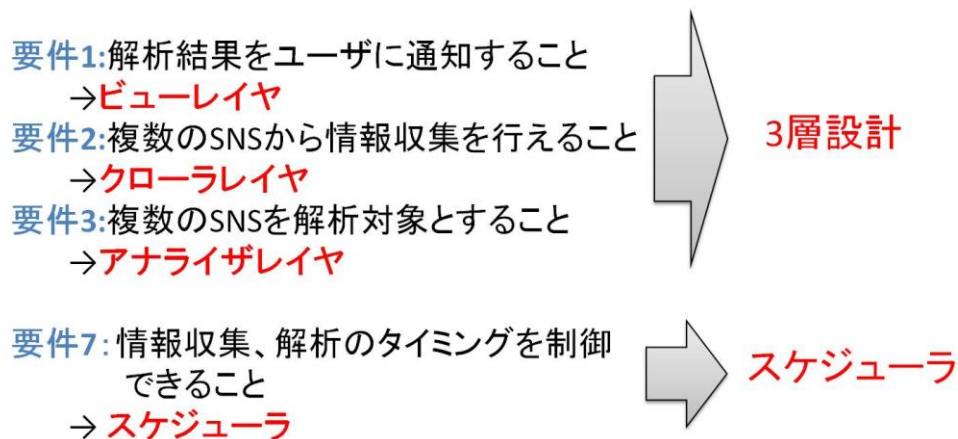


図 3-1 システム設計方針（3 層設計）

さらに、要件 4 と要件 5 を満たすために、図 3-2 ように各層の間のデータアクセス用インタフェースを定義した。要件 6 を満たすために、ユーザの SNS アカウントから取得したデータがどんな言語で記述されているかを判定する言語判定機能を設計した。

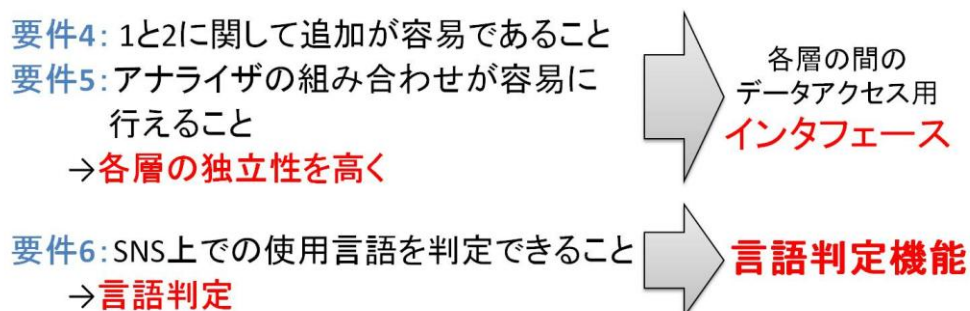


図 3-2 システム設計方針（スケジューラとインタフェースと言語判定機能）

上記の設計方針のアーキテクチャを図 3-3 に示す。

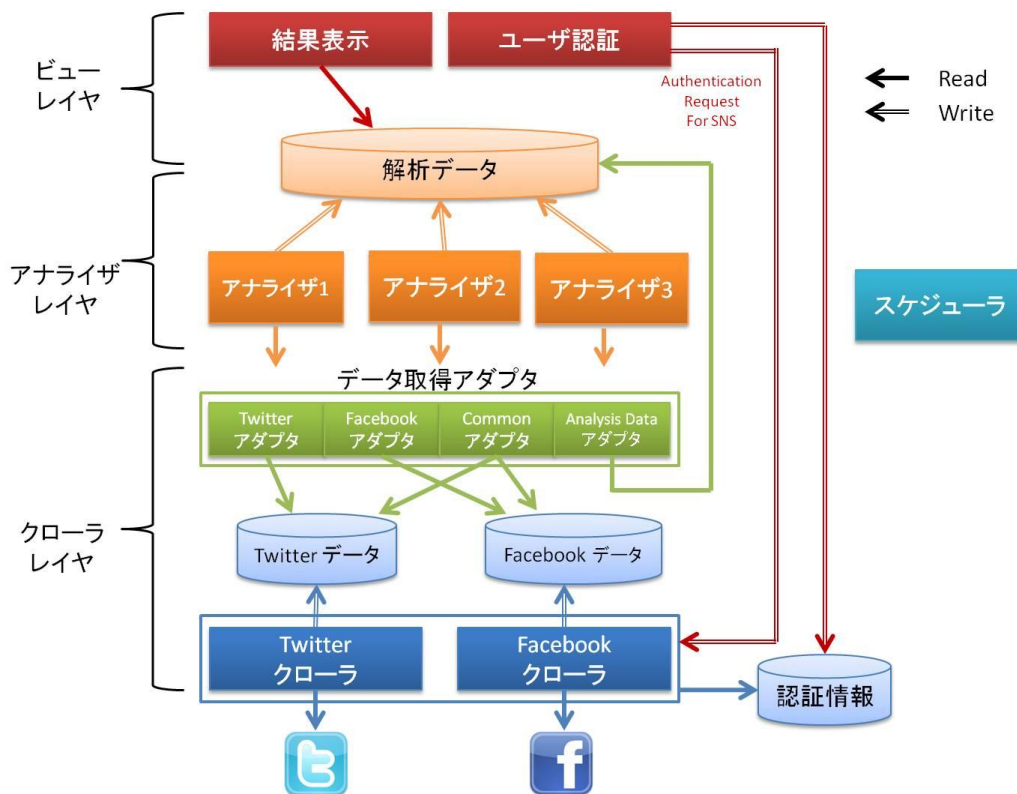


図 3-3 システムアーキテクチャ

図 3-3 において、クローラレイヤでは各 SNS からユーザ情報の収集を行い、また収集したデータを容易に取得できるアダプタを提供する。データ取得アダプタを利用することで、アナライザが Twitter のデータを使いたければ、その Twitter アダプタを利用すればよい。Twitter と Facebook 両方のデータを使いたければ、Common アダプタを使う。アナライザレイヤでは、アナライザを用いてクローラが収集したユーザ情報に対して解析を行う。ビューレイヤはユーザ情報収集のためのユーザ認証と解析結果の通知を行う。スケジュールラはクローラレイヤ、アナライザレイヤ及びビューレイヤの動作タイミングの制御を行う。

3.2 各レイヤの機能

3.2.1 クローラレイヤの機能

クローラレイヤでは各 SNS からユーザ情報の収集を行う。以下はクローラレイヤの機能である。

- ① SNS からユーザの情報を取得する機能
 - ユーザの認証済み SNS からユーザの情報を取得する。
- ② 解析対象となる SNS に拡張性を持たせる機能
 - 将来的に、Facebook と Twitter 以外の SNS を利用できるよう、拡張性を持たせる。
- ③ SNS 上での使用言語判定機能

- ユーザ情報の解析を行うために、言語依存のアナライザを選ぶ時、SNS 上でユーザが利用している言語を判定しなければならない。

3.2.2 アナライザレイヤの機能

アナライザレイヤではクローラが収集したユーザ情報に対して解析を行う。以下はアナライザレイヤの機能である。

① 人物属性推定機能

- Twitter API・Facebook API を用いて取得した情報を一部解析する。ユーザ情報の解析手法としては、日本語係り受け解析器である Cabocha 出典のような自然言語解析ツールを用いる。

② アナライザが 1 つ以上の SNS のデータを用いることができる機能

- 個人情報の抽出に用いるアナライザとしては、図 3-4 のように、ある SNS 単独のデータを用いたアナライザ、または図 3-5 のように、複数の SNS データを用いたアナライザの両方を提供できるようにする。



図 3-4 一つのアナライザが一つの SNS データを解析に用いる



図 3-5 一つのアナライザが複数の SNS データを解析に用いる

③ 他言語で開発されたアナライザを対応する機能

- 本システムが Python 言語で実装されるが、高精度な解析結果を得ることやアナライザの互換性のために、Python 以外の言語で実装されたアナライザを対応する必要がある。

3.2.3 ビューレイヤの機能

ビューレイヤはユーザ情報収集のためのユーザ認証と解析結果の通知を行う。以下はアナライザレイヤの機能である。

① ユーザ認証機能

- SNS から、API を用いたデータ取得のためのユーザ認証を行う必要である。

② 個人情報の通知機能

- 解析データを通知するインタフェースを実現する。具体的な通知情報を表 3-1 に示す。

表 3-1 SNS ユーザに通知する個人情報

個人情報	Twitter の解析結果	Facebook の解析結果	...
登録名	○	○	
居住地	○	○	
所属組織	—	○	
性別	△	△	
...			

3.2.1 スケジューラの機能

スケジューラはクローレイヤ、アナライザレイヤ及びビューレイヤの動作タイミングの制御を行う。以下はアナライザレイヤの機能である。

- ① SNS ユーザのデータ収集や情報解析の実行の順番を制御する機能
- ② 各レイヤ間の動作タイミングを制御する機能

3.3 システム動作

本システムのクローレイヤ、アナライザレイヤ、ビューレイヤの間では、互いの動作が独立で、各レイヤの間の動作連携はスケジューラを使用する。ユーザのログインから結果の表示までの間の各レイヤとスケジューラの動作を以下に示す。

- ① ユーザがシステムにログインする。
- ② ユーザのログイン情報をデータベースに保存する。
- ③ スケジューラがログインしたユーザに対して情報収集を着手する。
- ④ スケジューラがクローラにクローリング操作の要求をする。
- ⑤ クローラは SNS から情報を取得し、データベースに情報を保存する。
- ⑥ クローラはスケジューラにクローリング終了通知を行う。
- ⑦ スケジューラはアナライザを呼び出して、情報解析を行う。
- ⑧ アナライザはデータベースから SNS の情報を取得し、データ解析を行い、解析結果をデータベースに保存する。
- ⑨ 解析終了をスケジューラに通知する。
- ⑩ スケジューラがビューを呼び出す。
- ⑪ スケジューラが解析結果を画面に表示する。

3.4 設計の詳細

3.4.1 クローレイヤ

クローレイヤでは各 SNS からユーザ情報の収集を行う。以下はクローレイヤの設計・実装詳細について説明する。

1. ユーザ情報の取得

ユーザの利用している Facebook と Twitter の情報を取得するため、Twitter では、Python 言語のための TwitterAPI ラッパーライブラリである tweepy[7]を用い、Facebook では、Facebook が提供されている API[8]を用いて情報を取得する。

2. キューの使用

スケジューラは直接クローラに対してクローリング操作の要求を行う。スケジューラが同時に複数の要求を行えるよう、スケジューラと各クローラの間にはキューを導入する。スケジューラはクローラにクローリング操作の要求を行うとき、クローリング操作の要求をメッセージにしてキューに入れる（エンキュー）。クローラは、対応するキューからメッセージを取り出し（デキュー）、メッセージに含まれる取得する対象を SNS から取得する。また、要件 7 の「情報収集、解析のタイミングを制御できること」を実現するため、デキューの優先度が調整できる Q4m というメッセージキューソフトウェアを用いて、永続的な優先度付きキューを実装する。

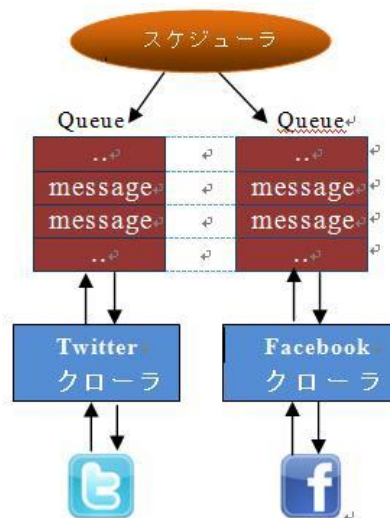


図 3-6 Queue の使用

3. データの格納

本プラットフォームでは MongoDB を用いて、SNS から収集したデータを保存する。

MongoDB はドキュメント指向型データベースである。MongoDB は 1 つのデータベースに複数のコレクションを持つ (MySQL でいうテーブル)。その各コレクションに多くのドキュメントを保存する。MongoDB に格納するデータの単位はドキュメントと呼ばれる、Twitter API で取得した JSON オブジェクトを (ほぼ) そのまま格納できる。また、ドキュ

メント指向データベースは従来のリレーショナル・データベースとかなりの違いがある。ドキュメント指向データベースは基本的に、スキーマを定義しなくても良い。SNS からのデータ構造の変更により、データ構造への対応が容易に実現できるようになっている。

4. アダプタ

Facebook や Twitter から取得し、データベースに保存したデータの形式は異なる。アナライザが保存したデータを容易に取得できるよう、アナライザとデータベースの間に、データベースをアクセスできるアダプタを導入する。アナライザはアダプタを通じてデータベースからデータを取得する。

また、単一の SNS 向けのアダプタと共用のアダプタを定義することにより、データ取得対象となる SNS の切り替えも容易にできるようになる。アナライザが Twitter のデータを利用する場合は、対応する Twitter アダプタを利用する。二つのデータを使いたければ、共用のアダプタを使えばよい。

3.4.2 アナライザレイヤ

本システムは Python 言語で実装されるが、高精度な解析結果を得ることやアナライザの互換性のために、Python 以外の言語で実装されたアナライザに対応する必要がある。そのため、アナライザレイヤを実装する際は、アナライザを作る必要となるアナライザの名前や概要、実行すべきメソッドが定義されたクラス名、アナライザが使用するデータや出力するデータ、またアナライザの実行頻度をすべて XML ファイルで定義する。XML ファイルに基づいて開発されたアナライザでさえあれば本システムで用いることができる。

解析結果を保存するためのデータベースは MySQL を使用する。ある解析項目に対してデータ形式が異なる解析結果を格納できるよう、解析結果のデータ構造をシリアライズし、バイト列としてデータベースに格納する。

3.4.3 ビューレイヤ

ビューレイヤはユーザ情報収集のためのユーザ認証と解析結果の通知を行う。以下はビューレイヤの設計・実装詳細について説明する。

1. 本システムのログインとユーザの認証

本システムを利用する際に、各 SNS の OAuth。SNS の OAuth 認証を用いる。本システムでは、パスワードの管理をしないし、ユーザとの間でパスワードのやり取りも行わないため、システムの安全性が高い。

また、SNS の API を利用するためには、OAuth 認証を必要とする。つまり、OAuth 認証が成功すると、本システムのログインと SNS の API 使用の許可を同時に得ることができる。

2. 解析結果の表示

本システムはアナライザが解析した結果を Web 画面でユーザに表示する。

具体的な方法は図 3-7 に示す。HTML と CSS を用いて静的なウェブ画面を開発し、オープンソースである Django の template モジュールを利用してアナライザから受け取った分析結果をウェブページの該当するところに埋め込む。

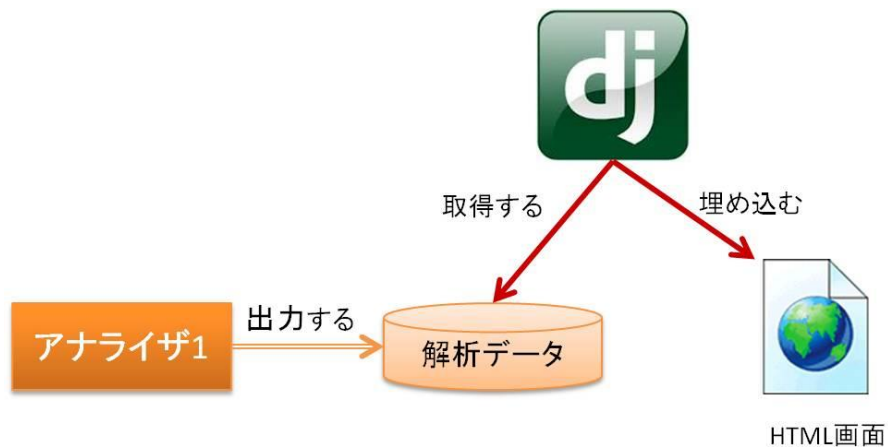


図 3-7 Django により結果表示

3.4.4 スケジューラ

スケジューラはシステム全体の流れをコントロールし、各レイヤ動作のタイミングを管理する。以下は、スケジューラが他のレイヤとの連携を説明する。

① ビューレイヤとの連携

- スケジューラは、ユーザの本システムへの登録およびログイン時にはログイン通知イベントを受け取って、クローラレイヤにクローリング操作の要求を行う。
- 解析完了後、解析結果をデータベースに保存する。

② クローラレイヤとの連携

- ユーザ認証済みの SNS に対してのクローリング操作の要求をクローラレイヤに送る。

③ アナライザレイヤとの連携

- クローリングしたデータ項目をアナライザレイヤのアナライザに送り、データの解析を開始させる。

第4章 アナライザ用周辺機能及び GUI の設計

本章では、筆者が担当したアナライザ用周辺機能及びシステム GUI 設計について述べる。4.1 節では、システム開発の作業分担について述べる。4.2 節では、アナライザ周辺機能である言語判定機能について述べる。4.3 節では、動的にクラスローディング機能について述べる。4.4 節では、システムの GUI について詳しく説明する。

4.1 開発作業の分担

今回のシステム開発はウォーターフォールモデルを 2 回反復する開発手法をとった。第 1 反復はすべての要件を満たすプラットフォームを実現し、単一 SNS のユーザ情報の解析を正確に行えることを目標とした。第 2 反復は解析対象となる SNS と脱匿名化アルゴリズムを検討し、より正確な解析結果を提示できるように改良する。

第 1 反復と第 2 反復で開発した機能とその担当者を表 4-1 に示す。なお、太字の部分は筆者の担当部分である。

表 4-1 開発作業の分担

第 1 反復	クローラレイヤ	クローラレイヤ本体の開発	長野
		アダプタの開発	長野
		言語判定機能の開発	孫
	アナライザレイヤ	アナライザレイヤ本体の開発	João Orui
		動的にクラスローディング機能の開発	孫
	ビューレイヤ	GUI 開発	孫
		ユーザ認証機能の開発	田中
		解析結果の表示機能の開発	田中
	スケジューラ	スケジューラの開発	João Orui
第 2 反復	アナライザレイヤ	性別推定アナライザの開発	孫
		誕生日判定アナライザの開発	João Orui
		写真から仲のいい友人判定アナライザの開発	João Orui
		友人の位置解析アナライザの開発	長野
		システムの第三者評価	田中

4.2 言語判定機能

4.2.1 言語判定機能の概要

本システムの中で、各 SNS のクローラによって収集した情報やつぶやきから、個人のプライバシー情報の解析を行うアナライザがいくつか存在している。

ただし、一方では、Twitter や Facebook など世界規模の SNS は、世界各国のユーザに使われているので、SNS での使用言語も様々な言語があると考えられる。また、図 4-1 で示したように、自分の SNS アカウントで、一つの言語だけ利用するユーザもいれば、複数の言語を利用するユーザもいる。そのため、アナライザは、クローラを通じて収集したつぶやきがどんな言語で記述されているかあらかじめ把握することはできない。



図 4-1 SNS での言語使用の例

また、個人のプライバシー情報の解析を行うアナライザのほとんどは特定言語の情報に対して解析を行う。つまり、アナライザは個人情報解析を行う前に、解析対象となる個人情報のデータがどんな言語で記述されているか把握しなければならない。そのため、ユーザの SNS アカウントから取得したつぶやきやコメント等のデータがどんな言語で記述されているかを判断するという言語判定機能が必要である。ここ機能は日本語や、英語、中国語、スペインなどの世界の主な言語を 60 種以上対応できる。

言語判定機能を利用する流れは、図 4-2 のように、ユーザのつぶやきを一個ずつ言語判定して、判定の結果によって、ユーザのつぶやきをその言語に対応するアナライザに渡す。

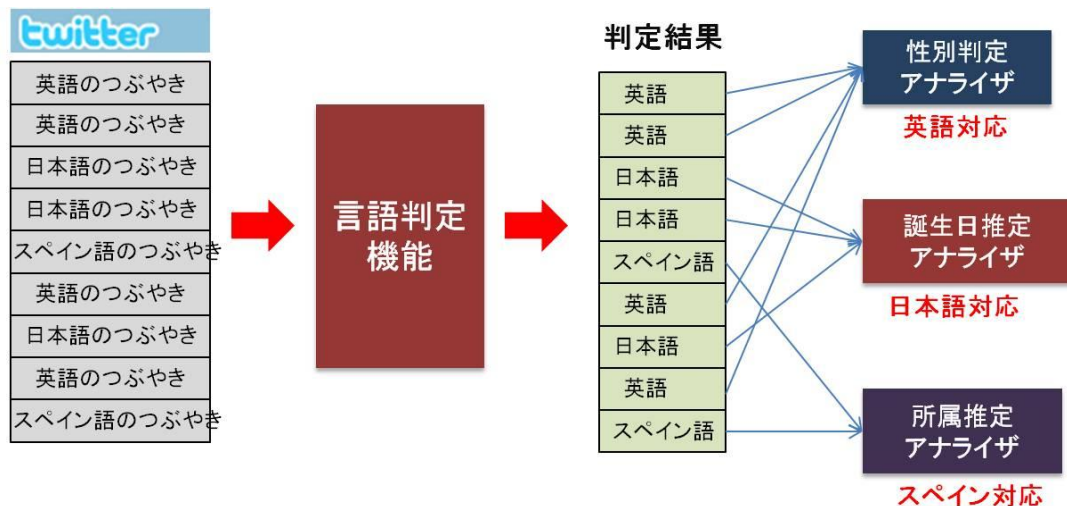


図 4-2 言語判定機能利用の流れ

4.2.2 既存ライブラリやサービスについて

言語判定機能の実現方法としては、仮に言語判定ライブラリを自ら作った場合、「対応言語数が少ない」ことや「判定精度が低い」等の問題があると考えられる。そのため、言語判定機能の実現には、既存の言語判定ライブラリやサービスを利用して言語判定機能を実現するという方法をとる。

表 4-2 は、既存の言語判定ライブラリとサービスを示したものである。

表 4-2 既存の言語判定ライブラリ・サービス

言語判定ライブラリとサービス	特徴
Lingua::LanguageGuesser[9]	■ 類似度ベースであるため、評価結果は相対値で得られる（判定範囲の 60 カ国語の中で、最も可能性の高い言語を一つだけ返す） ■ 短いテキストの判定精度が低い ■ 対応言語が多い
Guess Language[10]	■ 短いテキストの判定精度が低い ■ 判定しにくいテキストに対して、「未知」の結果を返す場合がある ■ 対応言語が多い
Google Language API[11]	■ オンラインアクセスのみ ■ 短いテキストでも比較的正確に判定できる ■ 対応言語が多い ■ 最も可能性の高い言語を一つだけ返す ■ 2011 年 12 月から無料で利用できなくなる

これらのライブラリやサービスは長いテキストや文章に対しての判定精度はかなり良いと掲載されている[12]。ただし、SNS には「短いつぶやきや短い文を書き込む」という特徴が

ある点を考えて、これらのライブラリやサービスが、SNS 上で短い文に対してどの程度の判定精度なのかを把握するために実験を行った。

● 実験の評価指標

ある言語判定ライブラリやサービスが本言語判定機能に十分な性能を持っているかどうかを以下の指標で評価した。

① 正解率

判定対象のつぶやきリストの中で、正しく言語と判定されたつぶやきの割合である。正解率が高くなると、このライブラリやサービスが本言語判定機能に適合すると言える。

② 誤判定率

与えられたつぶやきリストの中で、誤って判定されたつぶやきの割合である。誤判定率は1から正解率を引くことではない。理由は、正しく言語を判定できない場合は、「判定できない」や「未知」などの結果を返すことが期待されている。つぶやきが誤って判定された場合は、システムがこのつぶやきをこれに対応しないアナライザに渡してしまったので、アナライザの負担が大きくなる。そのため、誤判定率が低くなると良い。

③ 実行時間

与えられたつぶやきに対して言語判定にかかる時間である。現在、Facebook や Twitter 等の SNS 上では、1 万件以上のつぶやきやコメントを持っているユーザもいる。このようなユーザに言語判定する場合、判定にかかる時間がとても重要である。

● 実験の方法

具体的な実験方法は、まず、有名人等の Twitter アカウントから、日本語のつぶやき、英語のつぶやき、多言語を含むつぶやきを各 100 件収集する。Lingua::LanguageGuesser ライブラリと Guess Language ライブラリと Google Language API を用いてそれぞれ収集したつぶやきに対して言語判定を行う。

英語のつぶやきと日本語のつぶやきを実験の対象とする理由は、本システムは現時点で英語と日本語を使う SNS ユーザだけに対応させるという方針だからである。多言語を含むつぶやきを実験の対象とする理由は二つある。一つ目は、将来システムを拡張するとともに、システムの対応言語に新たな言語を追加する可能性がある。そのため、本言語判定機能は様々な言語の判定をできるべきであると考え。もう一つの理由は、多国語を話す人が自分の SNS アカウントで多言語を用いてつぶやきを発信するかもしれず、このようなユーザに対して言語判定の正解率を把握したいからである。

● 実験の結果

言語判定実験の正解率、誤判定率及び判定にかかった時間を図 4-3、図 4-4 と図 4-5 に示す。

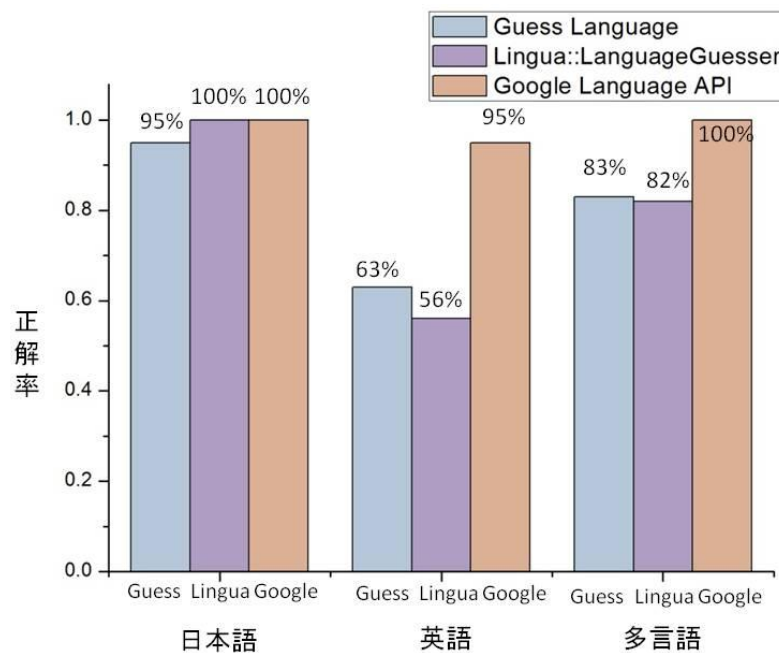


図 4-3 既存ライブラリとサービスの判定の正解率

図 4-3 に示すように、Google LanguageAPI で日本語、英語、多言語のつぶやきに対する言語判定で高い正解率を得られた。Guess Language と Lingua::LanguageGuesser の二つのライブラリは日本語のつぶやきに対しての判定正解率が高いが、英語のつぶやきを言語判定すると、それぞれ 63%と 56%という低い正解率となった。この結果を分析すると、SNS 上での英語のつぶやきは短いテキストが多く、英語やフランス語やドイツ語などのラテンアルファベット系言語の文法はお互いによく似ていることもあり、判定対象テキストが短い場合は、かなり判定しにくいと考えられる。

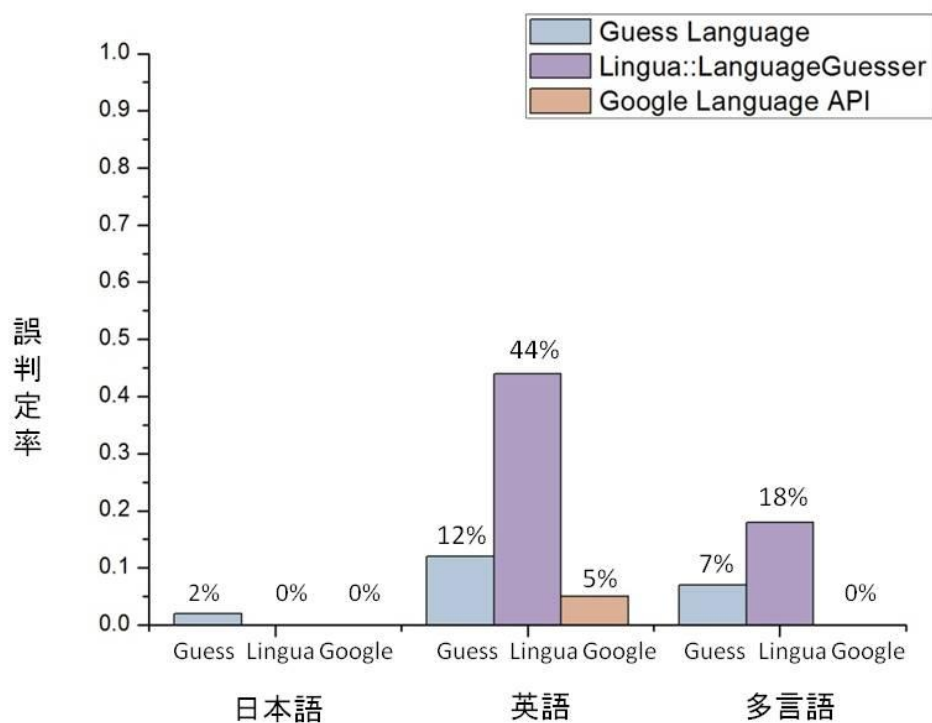


図 4-4 既存ライブラリとサービスの判定の誤判定率

図 4-4 に示すように、英語のつぶやきに対しての誤判定率は、Guess Language ライブラリが 12%の誤判定率に対して、Lingua::LanguageGuesser ライブラリは 44%の誤判定率と高くなった。この二つのライブラリは、図 4-3 に示すようにほぼ同じレベルの正解率を得られるが、誤判断率に大きな差がある。原因は、Guess Language ライブラリが判定しにくいテキストに対して、「未知」の結果を返すという特徴があるので、ラテンアルファベット系言語の短いテキストを判定しにくいテキストとして、「未知」の結果を返すからである。そのため、Guess Language ライブラリがラテンアルファベット系言語のテキストに対しての言語判定の誤判定率が低いという結果が見られた。

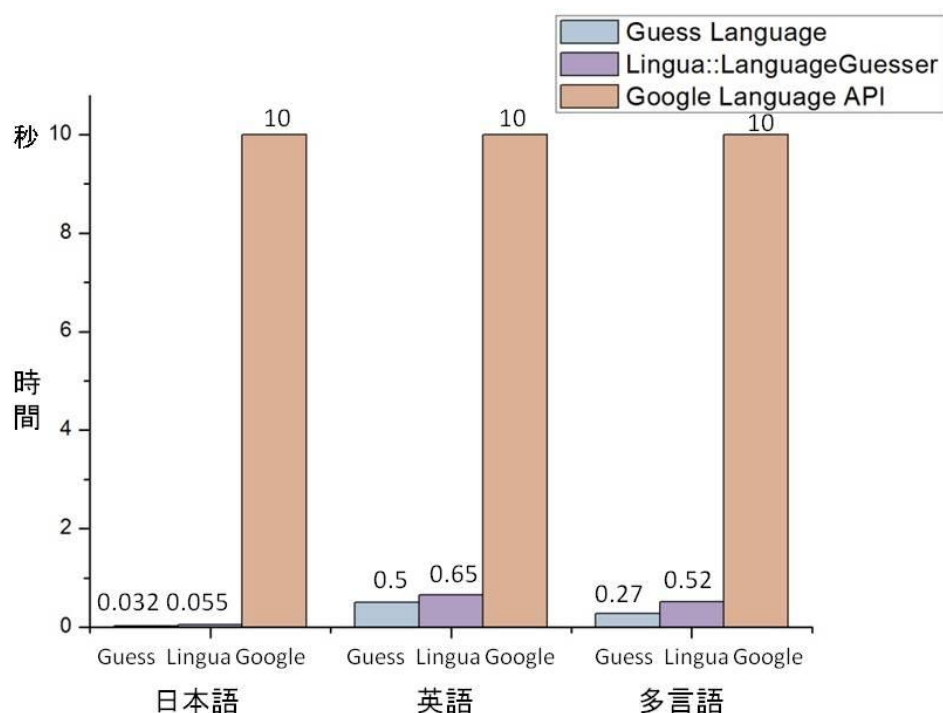


図 4-5 既存ライブラリとサービスが判定にかかった時間

言語判定にかかった時間について、図 4-5 に示す、GoogleLanguageAPI は、オンラインにアクセスするため、1 回のアクセスに 0.1 秒かかる。そのため、100 件のつぶやきを判定する場合、10 秒程度の時間がかかった。判定対象のユーザが数千件以上のつぶやきを持つ場合は、言語判定機能でかかった時間も数百秒になってしまった。これを踏まえて、GoogleLanguageAPI は判定の正解率が高いが、実行時間が長いため、本言語判定機能に用いるものとする。

以上の実験結果から、Guess Language と Lingua::LanguageGuesser 二つのライブラリを用いて言語判定機能を実現することとする。ただし、この二つのライブラリが英語やフランス語やドイツ語などのラテンアルファベット系言語のつぶやきに対しての判定正解率が低いという問題を解決しなければならない。次の節ではこの問題の解決方法について説明する。

4.2.3 言語判定機能の設計詳細

本節では、既存の言語判定ライブラリを利用して言語判定機能を実現する方法について述べる。

前節で Guess Language と Lingua::LanguageGuesser 二つのライブラリを用いて言語判定機能を実現することとした。ただし、この二つのライブラリが英語やフランス語やドイツ語などのラテンアルファベット系言語のつぶやきに対しての判定正解率が低いという問題がある。この問題を解決するために、二つのライブラリ及び SNS サービスに対して詳しく分析して、以下の特徴を発見した。

- Lingua::LanguageGuesser ライブラリは、判定対象となるテキストと、判定範囲の 60 ヶ国語との類似度をそれぞれ算出して、その中でいちばん類似度が高い言語を判定の結果として返すという形で言語判定を行っている。ソースコードの分析による、Lingua::LanguageGuesser ライブラリは言語判定を行う前に、動的に判定範囲を指

定できるという特徴があることが分かった。つまり、もともとの 60 カ国語の類似度比較範囲を任意の数カ国語に縮小でき、判定対象となるテキストと縮小した類似度比較範囲との間で類似度を算出することができる。

- **Guess Language** ライブラリは判定しにくいテキストに対して、「未知」の結果を返す。
- SNS のタイムラインでは、短いつぶやきと長いつぶやきは同じ言語で記述されている可能性が高い。

以上に述べた特徴を考えて、筆者は **Guess Language** と **Lingua::LanguageGuesser** ライブラリの組み合わせで言語判定を行うという方法を提案する。

具体的な方法としては、まず、**Guess Language** ライブラリを用いて与えられたつぶやきに対して 1 回目の言語判定を行う。**Guess Language** ライブラリは判定しにくいテキストに対して、「未知」の結果を返すという特徴があるため、1 回目の言語判定には、判定しやすいつぶやきだけに対して言語判定して、判定しにくいつぶやきを判定しない。次に、**Lingua::LanguageGuesser** ライブラリを利用して、1 回目の言語判定で判定できなかったつぶやきに対して 2 回目の言語判定を行う。ただし、2 回目の判定を行う前に、1 回目判定の結果によって 2 回目判定の類似度比較範囲を動的に指定する。1 回目判定で判定しなかったつぶやきを、指定の類似度比較範囲で判定していく。理由は、SNS のタイムラインが「短いつぶやきと長いつぶやきは同じ言語で記述されている可能性が高い」という特徴があるので、短いつぶやきに対しての類似度比較範囲を、長いつぶやきに対しての判定結果の範囲に限定すれば、判定の正解率が上がることが期待できると考えられる。

4.2.4 実験

次に、提案する言語判定方法の有効性を検証するため、二つのライブラリの組み合わせで前回の実験で収集したつぶやきに対してもう一度実験を行った。

図 4-6 は 100 件の英語のつぶやきに対しての言語判定の動作詳細を説明する。まず、**Guess Language** ライブラリを用いて 1 回目の言語判定を行う。1 回目の判定結果からみると、63 件が「英語」と正しく判定され、緑色で表示される。25 件が判定しにくい原因で判定されなく、灰色で表示される。他の 12 件が誤って他の言語と判定される。

それからの 2 回目の言語判定では、**Lingua::LanguageGuesser** ライブラリは 1 回目判定の結果によって、「英語、スペイン語、フランス語」という新たな類似度比較範囲を算出して、類似度の比較範囲をもともとの 60 カ国語から英語、スペイン語及びフランス語の 3 カ国語に縮小する。

次は提案する方法を用いて、多言語を含むつぶやきに対して言語判定を行う。図 4-8 に示すように、Lingua::LanguageGuesser ライブラリは、1 回目の判定結果が「未知」であるつぶやきに対して 2 回目判定を行う前に、動的に類似度比較範囲を英語、スペイン語、フランス語、スウェーデン語ドイツ語という 5 カ国語に縮小する。

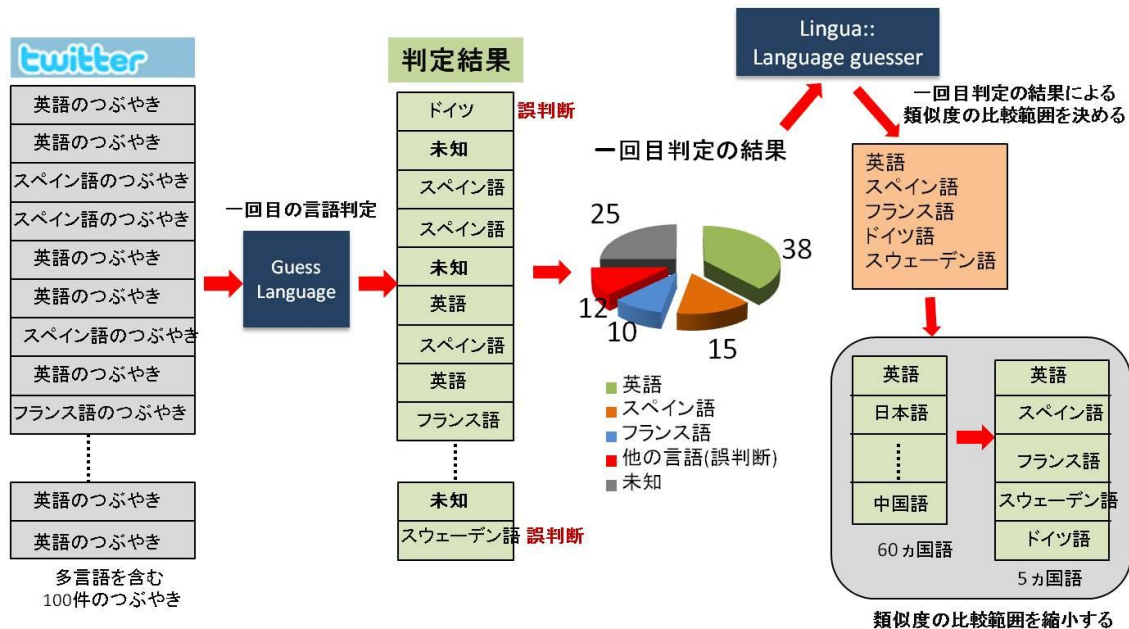


図 4-8 多言語を含むつぶやきに対しての言語判定の流れ

● 実験の結果

図 4-9 に提案する言語判定手法を用いて、英語のつぶやきや多言語を含むつぶやきに対して、言語判定した結果を示す。Guess Language と Lingua::LanguageGuesser ライブラリの組み合わせで英語のつぶやきと多言語を含むつぶやきに対して言語判定した結果、88%と 90%の正解率が得られた。二つの結果から単独ライブラリの言語判定より正解率が上がっていることが分かる。特に英語のつぶやきに対しての言語判定の正解率の大幅な向上が見られた。そのため、実験の結果を通じて、提案する Guess Language と Lingua::LanguageGuesser ライブラリの組み合わせの手法が SNS 上でのつぶやきやコメントに対しての言語判定に有効であると言える。

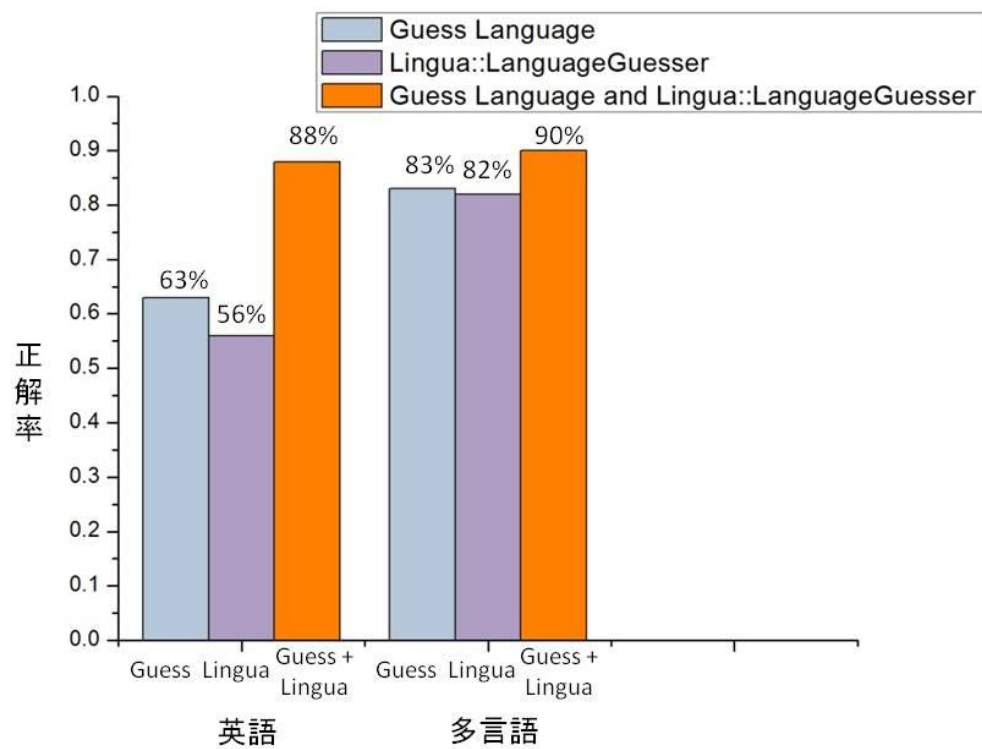


図 4-9 言語判定実験の結果

4.3 動的にクラスローディング機能

4.3.1 動的にクラスローディング機能の概要

1.2 節で記述している本システムの目的の一つは、SNS のプライバシー問題の解決を目指す研究の手助けとなるプラットフォームを作成するということである。この目的を実現するため、研究者の提案した解析アルゴリズム（アナライザ）を容易に応用できるプラットフォームを研究者に提供しなければならない。ただし、現在 SNS のプライバシー問題に携わっている研究者が提案した解析アルゴリズムを、本システムの開発言語である Python 以外の開発言語でも実装できることが望ましい。現在、Java、Python と Perl などのプログラミング言語がよく使われているため、今回のシステム開発では、これらの言語で実装された解析アルゴリズム（アナライザ）を本システム上で動作させることを目標とした。

本システムは Python 言語で記述されるため、Python 言語の `import` 文を用いて、Python 言語の解析アルゴリズム（アナライザ）の読み込みを行う方法は、モジュールの**静的読み込み**と呼ばれる。ただし、静的読み込みを利用して、新たな解析アルゴリズム（アナライザ）を本システムに追加する場合は、既存のコードを変更する必要があるという問題がある。この問題を解決するために、`import` 文を用いず、与えられたモジュール名を用いて、このモジュール名に対応するモジュールを動的に読み込む方法をとった。この方法はモジュールの**動的読み込み**と呼ばれる。筆者が担当した動的なクラスローディング機能は、モジュールの動的読み込みを用いて、Java や Perl などの Python 以外の言語で実装された解析アルゴリズム（アナライザ）を呼び出すことができるものである。

4.3.2 アナライザの実装

新たな解析アルゴリズムをアナライザとして本システムで利用するためには XML ファイルの作成と指定抽象クラスのサブクラス作成必要となる。

まず、アナライザに関する情報を定義する XML ファイルを作成する。この XML ファイルはアナライザことに定義され、ファイルの名前が「`manifest.xml`」とする。`manifest.xml`では下記の項目を定義する。

- 実装言語
- アナライザファイルのパス
- パッケージ名・クラス名
- アナライザ名
- アナライザの概要
- 入力項目及び出力項目
- 解析頻度
- アナライザ独自のレンダリング機能を有するか

システムは指定のディレクトリで `manifest.xml` ファイルを読み込んで、これらの項目を解析する。解析した項目によってアナライザを生成する。図 4-10 は `manifest.xml` ファイル定義の例である。

```

<analyzer>
  <class language="java" jarfile="path/sample.jar">java. Analyzer</class>
  <name>Java Analyzer1</name>
  <description>this is a test java analyzer</description>
  <scopes>
    <scope>twitter。 timeline</scope>
    <scope>twitter。 profile</scope>
  </scopes>
  <frequencies>
    <frequency>REGISTERD</frequency>
    <frequency>LOGGED_IN</frequency>
    <frequency>EVERY_WEEK</frequency>
  </frequencies>
  <OutputItems>
    <OutputItem>global。 name</OutputItem>
    <OutputItem>global。 birthday</OutputItem>
  </OutputItems>
  <renderable>False</renderable>
</analyzer>

```

図 4-10 manifest.xml ファイルの例

また、Java と Perl 言語を用いてアナライザを実装する際は、本システムが提供する抽象クラスである `AbstractJavaAnalyzer` と `AbstractPerlAnalyzer` を継承したサブクラスを実装する必要がある。そしてサブクラスで `invoke` と `getRenderView` という抽象メソッドを実装しなければならない。`AbstractJavaAnalyzer` と `AbstractPerlAnalyzer` 抽象クラスでは、解析を行うメソッドである `invoke` と、解析結果を返すメソッドである `getRenderView` のインタフェースのみを定義する。サブクラスがこの二つのメソッドを実装するだけで、本システムがアナライザを呼び出すことができるようになる。図 4-11 は `AbstractJavaAnalyzer` 抽象クラスのサンプルコードである。

```

public abstract class AnalyzerInterface {
    /**
     * アナライザの実行
     */
    public abstract void invoke(String user_id);

    /**
     * 解析結果表示の取得
     */
    public abstract String getRenderView(String user_id);
}

```

図 4-11 `AbstractJavaAnalyzer` クラスのサンプルコード

4.3.3 Java と Perl 言語への呼び出し

本節では、動的にクラスローディング機能が Java と Perl 言語の解析アルゴリズム（アナライザ）を呼び出す方法について述べる。

動的にクラスローディング機能を実現するために、Python 言語から Java や Perl 言語を呼び出す方法について調査した結果、比較的使いやすい方法としては、Python 言語の `subprocess` モジュールの `Popen` オブジェクトを使った方法があり[13]、これを用いることとした。`subprocess` モジュールが新しくプロセスを生成できるため、生成したプロセスから Java や Perl 言語のプログラムを呼び出すことで、Python 言語が Java や Perl 言語のプログラムをアクセスできるようになる。図 4-12 は具体的な呼び出し例を示す。

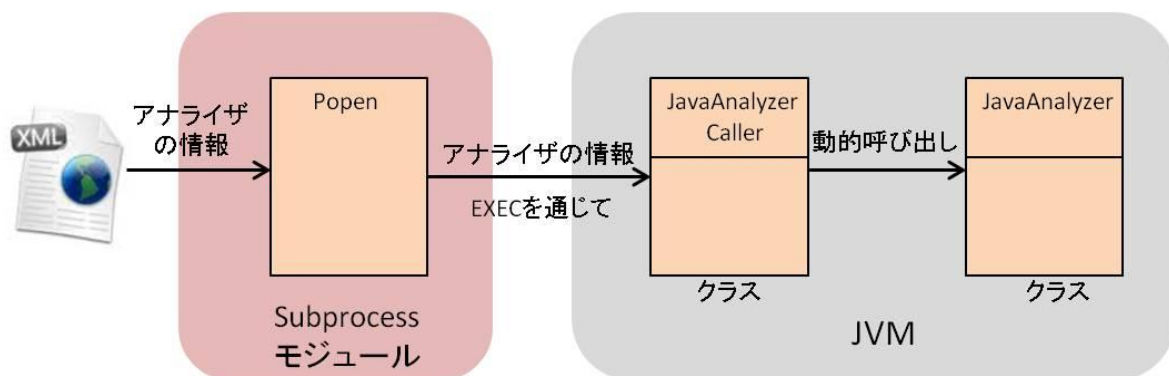


図 4-12 Python 言語から Java 言語のアナライザを呼び出す

`subprocess` モジュールは `Popen` オブジェクトを通じて子プロセスを生成する。この子プロセスが外部プログラムを実行するメソッドである `exec` を通じて用意した `JavaAnalyzerCaller` クラスを呼び出して、`manifest.xml` で記述するアナライザの `Jar` ファイルパス、クラス名及びユーザ ID をパラメータとして渡す。`JavaAnalyzerCaller` クラスは与えられた情報を用いてアナライザを動的に生成し、アナライザのメソッドを呼び出す。アナライザの呼び出しが終了と、`subprocess` モジュールの方が `exec` の標準入出力を通じて、`JavaAnalyzer` の出力を受け取る。`Perl` 言語で実装されたアナライザの呼び出す方法も `Java` の場合とほぼ同じである[14]。

4.4 GUI の開発

4.4.1 画面の設計

本システムは主にユーザ認証ページ・解析結果表示ページ（基本情報・詳細情報）の 3 ページがある。以下は各ページに対して説明を行う。

まず、ページの画面をデザインする際には、画面の配色を考えなければ見やすい画面を作ることができない。見やすい画面を作るには、背景と文字の明度・色相の差を大きくすれば良いと考えられる。また、画面設計において配色方法という研究[15]では、3 色を 70%:25%:5 くらいの比率にすると画面が見やすいと指摘している。それにより、今回のシステムの配色には、基本的に白い背景と赤い画像と黒い文字の組み合わせを用いる。色の比率を図 4-13 に示す。



図 4-13 画面配色の比率

4.4.2 画面の実装

毎画面は HTML 言語の DIV タグと CSS を用いて実装する[16]。DIV タグはレイアウト変更が容易やメンテナンス性が高いなどの利点があるため採用している[17]。また、他のレイヤと連携するため、Django 開発フレームワークを用いる。

4.4.3 ユーザ認証ページ

図 4-14 は本システムのユーザ認証ページである。



図 4-14 ユーザ認証ページの画像

ユーザ認証ページの画像には、システムのロゴやシステムの目標・説明や解析対象となる SNS の認証ボタンが含まれる。SNS 認証ボタンを押し、SNS 認証が完了すると次の解析結果通知のページに遷移する。

4.4.4 解析結果表示ページ

解析結果表示ページ（基本情報）の画像は図 4-15 の通りである。



個人情報
愚痴解析
年齢推定
活動時間分布
誕生日解析
などなど

プロフィール



名前 Name

 筑波 大郎
(from Analysis XXX)

 筑波 二郎
(from Analysis XXX)

 筑波 三郎
(from Analysis XXX)

住所 Address

 東京
(from Analysis XXX)

 茨城県
(from Analysis XXX)

 天久保2-1-1
(from Analysis XXX)

図 4-15 解析結果ページの画面（基本情報）

解析結果表示ページ（基本情報）の画面には、画面上部に表示されるシステムロゴと写真、画面左部のメニューと画面右部の解析結果表示部分が含まれる。解析結果表示部分では、主にユーザの名前や住所や所属などの基本情報を表示する。

図 4-15 の画面右部の基本情報表示部分では「名前」や「住所」や「生年月日」及び「所属」などいろいろな項目がある。毎項目の中では図 4-16 のように取得・解析した情報と取得先 SNS のロゴとアナライザを表示する。項目の表示範囲は情報の数により自動に延長・短縮ができる。

所属 University	
	筑波大学大学院システム情報工学研究科 (from Analysis XXX)
	大連交通大学ソフトウェア工学部 (from Analysis XXX)
	東京大学人文科学研究科 (from Analysis XXX)

図 4-16 解析結果表示部分

4.4.5 画面のテスト

近年では、IE、Firefox、Google Chrome など、複数の Web ブラウザが併用されている。これらのブラウザでは同じ HTML でも異なる表示になることがある。これに対応するため、画面テストを行う際には、各画面が以下の表の中のブラウザにおける正しく表示されるかを確認した。

表 4-3 画面テストで使うブラウザ

ブラウザ	バージョン
Internet Exploer	7、8、9
Firefox	4、5、6
Google Chrome	14.0

また、画面の解像度が異なる端末に、間違う表示に対応するため、以下の表の中の解像度における正しく表示されるかを確認した。

表 4-4 画面テストで使う解像度(ピクセル)

800 * 600	1024 * 768
1280 * 800	1280 * 1240
1600 * 1000	1600 * 1200
1920 * 1200	

第5章 性別推定アナライザの開発

本章では、筆者が開発した性別推定アナライザについて述べる。5.1 節では、アナライザを開発する際の役割分担について述べる。5.2 節では、性別推定アナライザの概要について述べる。5.3 節では、性別推定の関連研究について述べる。

5.1 アナライザの開発について

今回のシステム開発はウォーターフォールモデルを 2 回反復する開発手法をとる。その中で、第 2 回反復では、個人情報の解析アルゴリズム（アナライザ）をいくつか開発することで、どんな個人情報が抽出される可能性があるかをユーザに通知する。

そのため、第 2 回反復では、我々が自分の能力や興味に合わせて表 5-1 に示すようなアナライザを提案して開発した。太字の部分は筆者が担当した部分である。

表 5-1 アナライザ開発担当の分担

アナライザ名	担当者
性別推定アナライザ	孫
誕生日判定アナライザ	Joao
写真ベストフレンド判定アナライザ	Joao
友人の位置解析アナライザ	長野

次の節から、筆者が担当した性別推定アナライザについて説明する。

5.2 性別推定アナライザの概要

Twitter や Facebook などの SNS サービスでは、実名登録が求められ、いくつかの個人情報を登録することが求められるが、自分の実名を明かさなく、自分の個人情報をネットに書き込まないために、匿名登録や SNS が提供するプライバシー設定機能を用いて個人情報を隠すユーザもいる。ただし、悪意のある人は SNS ユーザの発言や友人関係や投稿した写真などを利用してユーザの個人情報や身元情報を推測し、不正な利用を行うという危険性がある。我々はユーザに対して性別、誕生日のような人物の属性を解析するアナライザを開発して、アナライザの解析結果をユーザに通知することで、ユーザに SNS の利用に対する注意喚起をしたいと考える。これらのアナライザの中で、筆者は、ユーザの SNS での発言からユーザの性別を推定する性別推定アナライザを開発した。

性別推定アナライザの推定対象は、日本語で発言する SNS ユーザである。また、性別推定の要素については、友人関係や活躍時間帯などの情報からユーザの性別を推定できるかもしれないが、そうすると、アナライザ設計における複雑さが増し、開発工数も膨大になる。筆者はユーザの SNS 上での発言を利用するだけでも、ユーザの性別を推定できると考え、日本語言葉の特徴を利用し、ユーザの性別を推定することとした。

5.3 関連研究

文章の著者に対する性別推定としては、平均 34000 語程度の英語の文章に対して、この文章の著者の性別を推定する研究[18]と、日本語のブログに対してブログの著者の性別を推定する研究[19]が行われている。正解率には、前者は 80%程度、後者は 85%以上である。

本アナライザとの相違点は 2 つある。一つは、前者は英語でなされたものであるのに対し、本アナライザは日本語のテキストを対象としている。もう一つは、後者の推定対象がブログの著者であり、本アナライザの推定対象が SNS ユーザであるという点である。それは、ブログの文章は文語的に記述されることが多く、SNS 上での発言は口語的に記述されることが多いという相違点があると考えられる。

5.4 性別推定の手法

SNS 上での発言は口語的に記述されることが多いという特徴がある。そのため、筆者は、日本語の話し言葉の特徴を考えて、以下の表現に着目することとした。

代名詞: SNS におけるユーザの発言や投稿では、ユーザ自身の考えや経験を述べることと、友達の投稿にコメントすることが多い。その場合、一人称代名詞はユーザが自分を指し、二人称代名詞はユーザの友達を指す際に使用しているものであると考えられる。現代日本語の話し言葉では、代名詞の使用における男女の差がある。たとえば、女性は 1 人称の「あたし」、2 人称の「あなた」をよく使い、男性は 1 人称の「おれ」、「ぼく」、2 人称の「おまえ」などをよく使う。また、「あいつ」、「こいつ」、「やつ」などの 3 人称の使用における男女の差もある。そのため、筆者は人称代名詞の 1 人称と 2 人称 3 人称全部が SNS ユーザの性別推定に有効であると考ええる。

終助詞: 話し言葉の男女差という研究[20]では、男女の話し言葉には、終助詞の使用において差があると指摘している。たとえば、女性は「ね」、「よね」、「のよ」、「かな」をよく使い、男性は「な」、「ぞ」、「ぜ」、「かい」などの終助詞をよく使う。以上にも述べた通り、SNS 上での発言は口語的に記述されることが多いと考えられるので、我々は終助詞が SNS ユーザの性別推定に有効であると考ええる。

ブログの著者の性別推定の研究では全ての「形態素」を利用しているが、今回開発する性別推定アナライザでは、開発時間を減らすため、「人称代名詞」、「終助詞」という二つの要素のみを利用する。

推定手法については、ブログの著者の性別推定の研究では、ブログの著者を“男性”、“女性”、“性別不明”の三つのクラスに分類するが、今回開発した性別推定アナライザでは、図 5-1 のように、確信度の比較を通じて、推定対象となる SNS ユーザを“男性”と“女性”という二つのクラスだけに分類する。確信度とは、与えられたユーザの発言を、男性と女性とそれぞれに判定する強さを計算した値である。

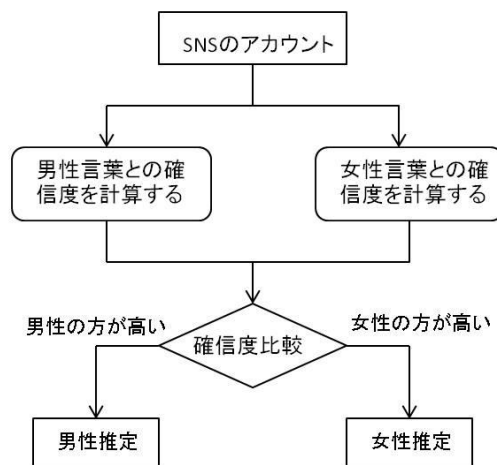


図 5-1 性別推定の手法

確信度計算については、図 5-2 のように、ユーザの SNS アカウントから取得した発言の中で、出現した代名詞と終助詞に、それに対応する確信値と出現回数を掛けるというアルゴリズムを用いる。代名詞と終助詞の確信値とは、この詞が男性っぽい言葉であるか、女性っぽい言葉であるかを推定する確率値である。たとえば、「僕」という代名詞の男性確信値が女性確信値より高く、また、「あたし」という代名詞の女性確信値の方が高い。

$$\begin{aligned}
 & \text{代名詞1} * \text{代名詞1の確信値} * \text{回数} + \text{代名詞2} * \text{代名詞2の確信値} * \text{回数} + \dots + \\
 & \quad + \\
 & \text{終助詞1} * \text{終助詞1の確信値} * \text{回数} + \text{終助詞2} * \text{終助詞2の確信値} * \text{回数} + \dots + \\
 & \quad \Downarrow \\
 & \text{確信度}
 \end{aligned}$$

図 5-2 確信度計算のアルゴリズム

代名詞と終助詞の確信値を計算するために、代名詞と終助詞毎に SNS 上で使われる頻度を把握しなければならない。このため、以下の実験を行った。

● SNS 上での話し言葉の男女差の実験

実験のデータについては、Twitter では、アカウントのプロフィール欄で自分の性別や名前や年齢などの個人情報などを記述しているユーザが存在する。このようなユーザはその情報をもとに性別を判断できる。また、プロフィール欄で性別や名前などを記述していないユーザについても、プロフィール写真に自分の写真を設定しているユーザは、その写真により性別を判断できる。

筆者が上述のような性別が判断できるユーザの Twitter アカウントから、男女各 50 名、合

計 15000 件のつぶやきを収集した。それから、日本語の形態素解析器である MeCab を用いて、収集したつぶやきに対して一件ずつ解析して、出現した代名詞と終助詞を取得した。表 5-2 は実験結果のまとめである。

表 5-2 SNS での代名詞と終助詞の出現頻度

つぶやきの件数：男性（7524）、女性（7339）			
代名詞（使用回数）		終助詞（使用回数）	
男性	女性	男性	女性
俺、257	私、538	か、985	ね、1025
これ、189	それ、179	ね、583	か、941
それ、149	これ、173	よ、507	よ、724
僕、113	何、169	な、469	な、334
私、92	みんな、147	なあ、425	よね、301
みんな、90	誰、95	かな、220	かな、237
誰、59	わたし、90	わ、194	わ、170
やつ、42	あたし、86	よね、136	よ一、161
あれ、35	こちら、71	よ一、99	の、124
こちら、26	あなた、65	ね一、73	なあ、103
彼女、25	やつ、47	ぞ、65	ね一、87
みなさん、25	あれ、40	な一、60	な一、85
君、23	彼、22	の、60	かなあ、49
あなた、21	皆、21	ぜ、52	ぞ、47
おれ、19	彼女、20	さ、35	ねえ、43
彼、19	君、17	ねえ、31	さ、39
お前、18	みなさん、15	べ、28	なあ、29
皆、15	彼ら、10		

使用回数からみると、上記表の中にある詞は大きく分けて 2 種類がある。

- 男性専用の単語と女性専用の詞。たとえば、'俺'、'僕'、'あたし'等の代名詞。
- 男女両方使用、使用頻度が違う単語。たとえば、'私'、'あなた'などの代名詞、'か'、'よ'、'ね'などの終助詞。

両方とも性別推定において使えるが、前者の重要性が後者より高いと考えている。その原因はたとえば次のような場面がある。SNS ユーザが自分の投稿内容の中で、'俺'、'僕'を用いた場合、女性は'俺'、'僕'という 1 人称代名詞をほぼ使用しないので、このユーザの性別が男性である可能性が非常に高いと言える。ただし、男性より女性の方が'ね'、'よ'をよく使用するが、このユーザの性別を確定的には判定できない。理由は男女にかかわらず、'ね'、'よ'などの終助詞をよく使用するからである。そのため、確信値を計算する際には、前者の確信値の大きさを高め、後者の確信値の大きさを下げるという方針で確信値を計算すべきである。

具体的な確信値の計算は、表 5-2 中の代名詞や終助詞の使用回数が男女合わせた使用回数の割合を確信値とする。ただし、回数をそのまま用いると学習データのノイズが直接反映し

てしまうため、使用回数に対して $1/N$ 乗を行い、ノイズを押えた。

N がどのような値を取ればより良い結果を得られるかを確かめるため、我々は N に異なる値を与え、得られた確信値を元に、前節で述べた男女合計 100 名の SNS ユーザに対して性別推定の実験を行った。実験の評価尺度には全 100 名の SNS ユーザのうちの正解したユーザの割合を示す正解率を用いた。実験の結果は図 5-3 に示す。

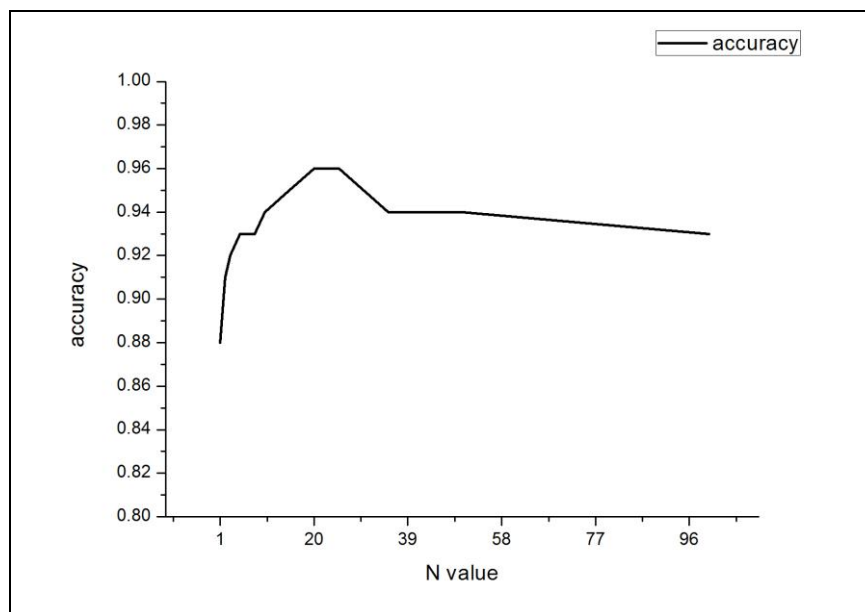


図 5-3 性別推定の実験の結果

N の値は 1 を取るとき、最低の正解率（88％）を得られた。N の値が大きくなると正解率が高くなる。N の値が 20 から 25 までの値を取るとき、最高の正解率（96％）を得られた。表 5-3 は N の値が 20 を取るとき算出した各代名詞と終助詞の確信値を示す。

表 5-3 性別推定の確信値（抜粋）

代名詞の確信値（男性：女性）	終助詞の確信値（男性：女性）
'私' : [0.413、0.587]	'か' : [0.502、0.498]
'お前' : [0.547、0.453]	'ね' : [0.472、0.528]
'あいつ' : [0.569、0.431]	'よ' : [0.482、0.518]
'俺' : [1.000、0.000]	'な' : [0.517、0.483]
'僕' : [1.000、0.000]	'なあ' : [0.570、0.430]
'わたし' : [0.000、1.000]	'かな' : [0.496、0.504]
'やつ' : [0.472、0.528]	'ぞ' : [0.516、0.484]
'あたし' : [0.000、1.000]	'よね' : [0.460、0.540]
'あなた' : [0.444、0.556]	'わ' : [0.507、0.493]
'彼ら' : [0.440、0.560]	'ぜ' : [0.537、0.463]
'オレ' : [1.000、0.000]	'ねー' : [0.459、0.541]
'皆' : [0.483、0.517]	'の' : [0.464、0.536]
'僕ら' : [1.000、0.000]	'ん' : [0.445、0.555]
'君' : [0.515、0.485]	'ネ' : [0.387、0.613]

今回の開発では、上記表中の確信値を用いて性別推定アナライザを実装した。

5.5 実験

提案した性別推定手法の有効性を確認するため、新たなデータで性別推定の実験を行った。

まず、実験の際に使用するデータについては、筆者が前節で述べたような性別が判断できるユーザの Twitter アカウントから、さらに男女各 25 名、每人平均 150 件のつぶやきを収集した。これらのデータは実験のデータとして使用する。

それから、開発した性別推定アナライザを用いて、収集したデータに対してユーザの性別推定の実験を行った。実験の評価尺度には全 50 名の SNS ユーザのうちの正解したユーザの割合を示す正解率を用いた。実験の結果は図 5-4 に示す。

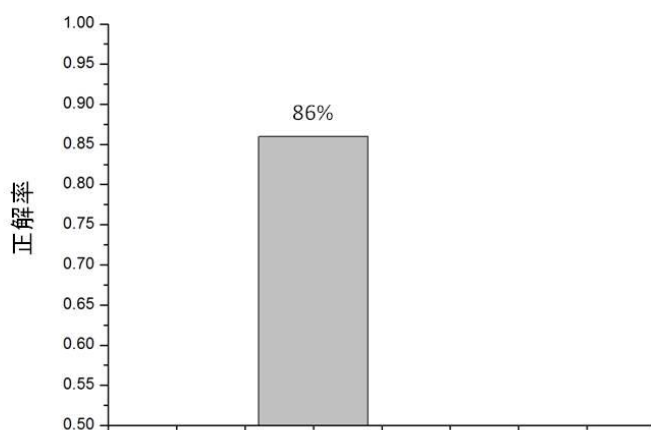


図 5-4 性別推定アナライザの正解率実験の結果

今回の実験では、86%の正解率を得られた。上述の実験を通じて、今回提案する性別推定の手法が SNS ユーザの性別推定に有効であると言える。

5.6 今後の課題

前節で行った性別推定アナライザの正解率実験では、86%の正解率を得られたが、この実験で利用する実験データの数は多くないので、この結果は最終の性別推定アナライザの正解率と言えない。今後はより多くの SNS ユーザの発言を収集して、これらの発言を用いて、より精度の高い正解率を求める。

第6章 プロジェクトの経過と進め方

6.1 プロジェクトの進め方

プロジェクトの企画、要件定義及び基本設計工程では、チームのメンバー全員でディスカッションする形を取った。システム設計と実装工程では、メンバーの一人で作業し、他のメンバーがレビューする方法を用いた。

作業を行う上で、各メンバーのタスクや進捗を管理するために、プロジェクト管理ツールである Redmine を用いる。また、開発の進捗などを指導教員に定期的に報告を行った。

6.2 開発スケジュール

今回のシステム開発では、我々はまずユーザの情報収集、データ保存、結果表示及び個人情報推定アルゴリズム（アナライザ）を容易にインストールできるプラットフォームの開発に重点を置いて、その後時間が許せる範囲でできるだけ多くの個人情報推定アルゴリズム（アナライザ）を増やす、という計画を立てた。そのため、今回の開発はウォーターフォールモデルを2回反復する開発手法をとった。第1反復はすべての要件を満たすプラットフォームを実現し、単一 SNS のユーザ情報の解析を正確に行えることを目標とした。第2反復は解析対象となる SNS とユーザ属性推定アルゴリズムを検討し、より正確な解析結果を求める。

6.2.1 開発スケジュールの計画

図 6-1 は本システムの開発スケジュールの計画を示す。

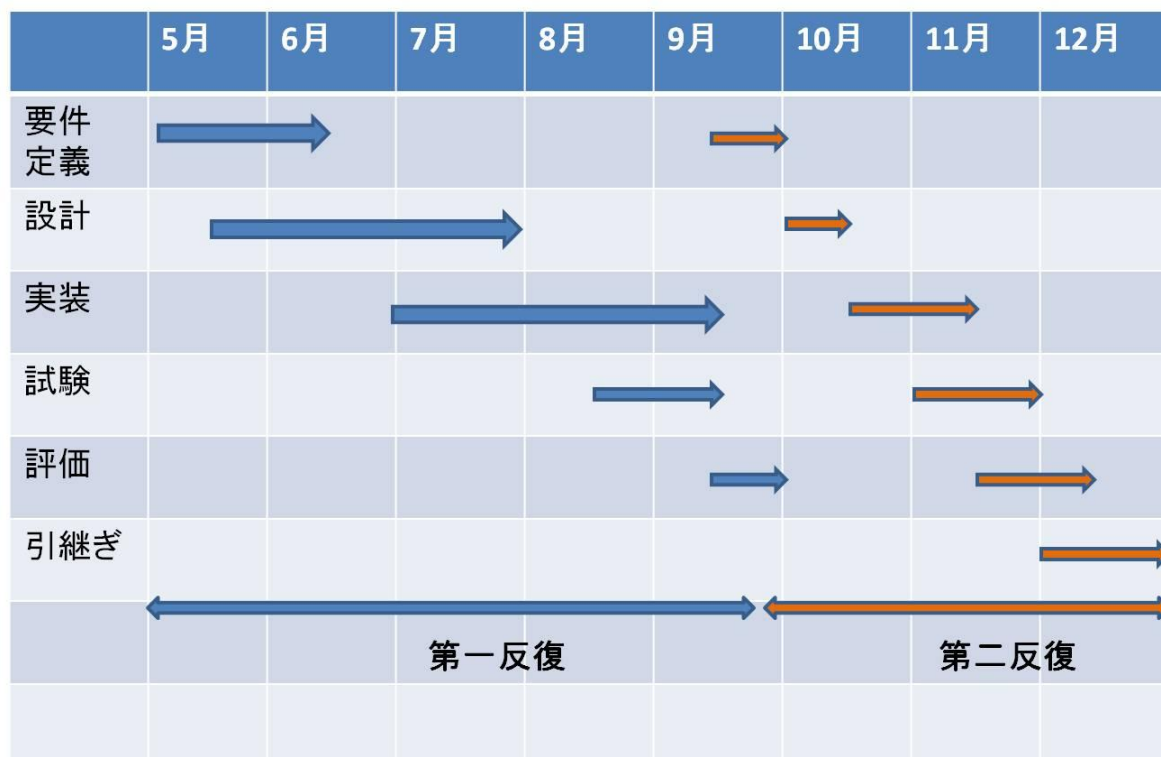


図 6-1 開発スケジュールの計画

第1反復については、まず、要件定義工程では、課題を洗い出し、課題解決へのアプローチを考え、テーマ指導先生の意見を頂き、システムの要件を定義した。設計工程では、要件定義工程で定義した要件を満たすことができるようにシステム設計を行い、システムを機能によって分割し、各機能の仕様を明確にした。また、ユーザに表示する画面のプロトタイプを作成した。実装工程では、開発メンバーが個別に実装を行い、システム要件を満たす機能を実装した。試験工程では、開発メンバーが自分の担当した機能に対して、単体・結合試験を行い、お互いにレビューを行った。最後の評価工程では、主観的にシステム要件の満足状況を評価方法として、第1反復で作成したシステムに対して評価を行った。

第2反復については、要件定義では、まず、どのようなアナライザが必要であるかの調査を行い、メンバーの興味と技術能力から一つあるいはいくつかのアナライザをメンバーに割り当てた。その後の設計実装工程では各メンバーが割り当てられたアナライザに対して開発を行った。最後に作ったシステムに対する評価した。

6.2.2 開発スケジュールの実績

本システム開発スケジュールの実績を図6-2に示す。各工程では、上の線は第1反復の計画開発スケジュールを、橙線は第2反復の計画開発スケジュールを、赤線は実績の開発スケジュールを示している。

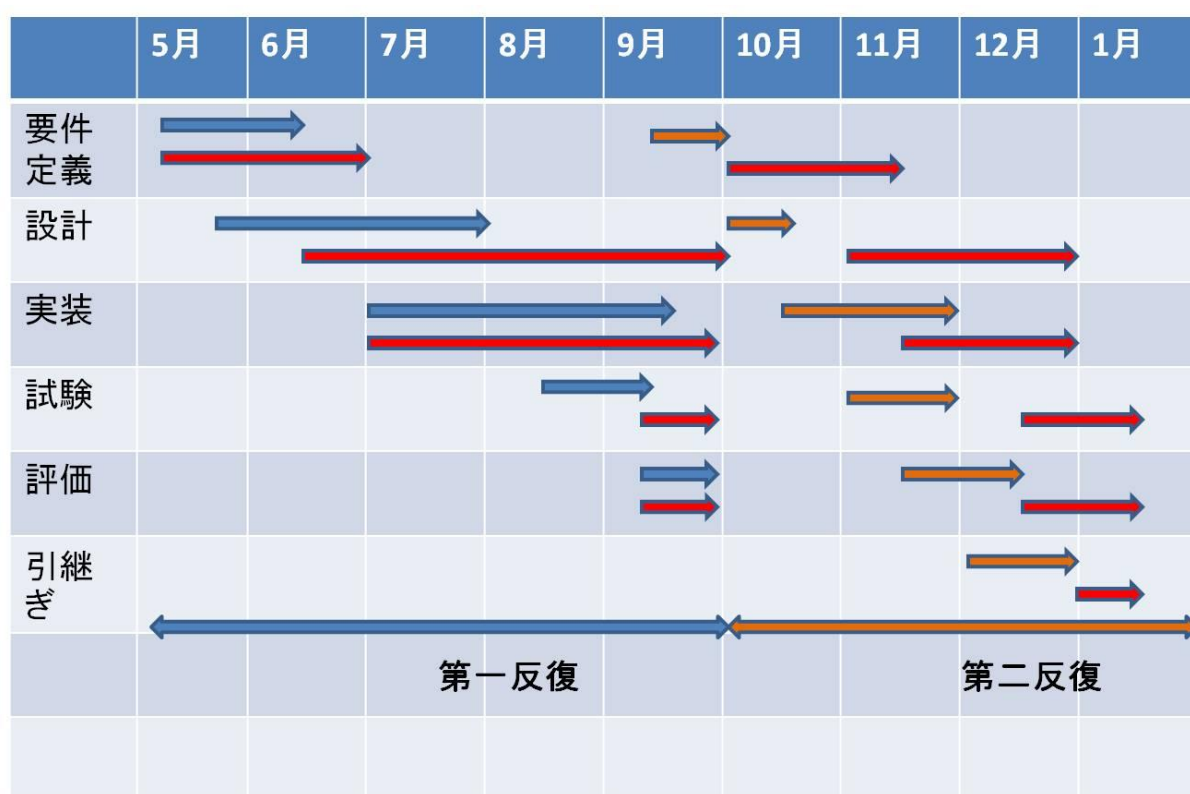


図 6-2 開発スケジュールの実績

図6-2から、第1反復では設計と実装工程の実績が計画より大幅に遅れていることが分かる。その原因は、筆者がシステム設計の経験不足で、設計を何度か繰り返し、スケジュールが設計工程で遅延してしまった。この問題点を反省した結果、第2反復では、担当した機能を開発する際に、未知なる技術に対する先行技術調査をしておくことで、開発作業が期間内に終わることができた。

第7章 システムの評価

本章では、本プロジェクトで作成したシステムに対して評価を行う。システムの評価は開発メンバ自身の観点と第三者の観点の両面から行う。7.1 節では開発者自身の評価について述べる。7.2 節では、第三者評価について述べる。

7.1 開発メンバによるシステムの評価

開発メンバによるシステムの評価には、筆者の視点で作成したシステムの要件満足状況を評価方法として評価を行った。

- 要件 1 : 解析結果をユーザに通知すること。
ユーザに表示する画面と解析機能を作成したので、この要件を満たしたと言える。
- 要件 2 : 複数の SNS から情報収集を行えること。
Facebook・Twitter それぞれに対応したクローラを作成し、この要件を満たしたと言える。
- 要件 3 : 複数の SNS を解析対象とすること。
Facebook・Twitter それぞれに対応したアナライザを作成したので、この要件を満たしたと言える。
- 要件 4 : 2 と 3 に関して追加が容易であること。
Facebook と Twitter に対応した情報収集を行うクローラを作成したことと、クローラとアナライザの間に独立性を確保する設計を行ったため、新たな SNS から情報収集する機能を容易に実装できる。また、Python 以外の言語で作成されたアナライザが本システムに実装できるよう設計を行い、新たなアナライザの追加が可能であるため、この要件を満たしたと言える。
- 要件 5 : アナライザの組み合わせが容易に行えること。
スケジュールの遅延やこの要件の優先度が低いため、この機能はまだ実現していないため、この要件を満たしていない。
- 要件 6 : SNS 上での使用言語を判定できること。
言語判定機能を実現するため、この要件を満たしたと言える。
- 要件 7 : 情報収集、解析のタイミングを制御できること。
スケジューラはユーザに対して優先的に情報収集を行い、クローラやアナライザのタイミングを調整する機能を有するため、この要件を満たしたと言える。

7.2 第三者によるシステムの評価

7.2.1 評価の目的

1.2 節で述べたように、「SNS 上での個人情報漏えい状況をユーザに通知する」と「SNS プライバシーに関する研究者の手助けとなるプラットフォームを作成する」という 2 点を実現することが本プロジェクトの目的である。そのため、システム評価の目的は、作成したシステムが、2 つの目的通りに機能しているかどうかを客観的に評価した。

7.2.2 評価の方法

まず、第三者に本システムを利用してもらった。その後、システムの利用に関するアンケートに回答してもらうことにより、作成したシステムを客観的に評価した。システム全体と、筆者が担当した部分の妥当性に関する評価項目に表 7-1 に示す。

表 7-1 アンケートの質問項目の一部

評価対象	質問番号	質問内容	記入方法
GUI の見やすさ	質問 4-2	本システムのデザインは見やすいものでしたか？	4 段階評価
GUI の使いやすさ	質問 4-4	全体的に評価して、本システムの操作感はどうでしたか？	4 段階評価
性別推定アナライザの正解率	質問 7-2	本システムが解析したあなたの個人情報について、それぞれ該当するものを選んでください。 - 性別	4 段階評価
システム全体	質問 11-1	本システム全体を通して、満足いただけるサービスだと感じましたか？	4 段階評価

7.2.3 評価の結果

61 名の被験者は本システムを使用してアンケートを回答した。質問 4-2 を回答した被験者では、システムのデザインが「見やすい」と「比較的に見やすい」が 9 割を占めていた。質問 4-4 を回答した被験者では、システムの操作感「操作しやすい」と「比較的に操作しやすい」が 8.5 割を占めていた。性別推定アナライザを使用した被験者では、推定結果が「完全に一致した」が 8.4 割を占めていた。システム全体の総合評価では、「満足できる」と「比較的に満足できる」が 6.4 割を占めていた。これらの評価結果により、おおむね当初の目的は達成できたと考ええる。

第8章 おわりに

本プロジェクトは、SNS プライバシー問題に対する SNS の利用者に注意の喚起をしたいという目的で、SNS からの意図しない個人情報漏えい検出システムの提案、開発、評価を行った。

システムの計画段階で、SNS プライバシー問題の原因を分析して検討した結果、SNS からどのくらい個人情報が漏れているかをユーザに通知する Web サービスを作成するというアプローチをとった。また、システムの要件を決定する際に、SNS のプライバシー問題の解決を目指す研究の手助けとなるように、本システムの柔軟性や拡張性の向上について検討を行った。

開発工程で、筆者が開発を担当した部分については、全体として見通し良く行うことができたが、第 1 反復で、システム設計の経験不足が原因で、設計を何度か繰り返し、スケジュールが設計工程で遅延してしまったという問題が発生した。この問題点を反省した結果、第 2 反復では、担当した機能を開発する際に、未知なる技術に対する先行技術調査をしておくことで、開発作業を期間内に終わらせることができた。

本プロジェクトで開発したシステムは、自分の SNS アカウントの個人情報の漏えい状況に関心を持っている SNS ユーザはもちろん、SNS のプライバシー問題に関する研究を行っている研究者にとっても研究活動の役に立つことが期待される。

謝辞

本プロジェクトを行うにあたり、課題担当教員である佐久間淳准教授と山本幹雄教授には、貴重なご意見やご助言を頂きました。深く感謝申し上げます。

指導教員である田中教授には、2年間懇切丁寧な御指導を頂きました。私自身、とても学び多い留学生生活を過ごすことができましたことを心より感謝いたしております。

「高度 IT 人材育成のための実践的ソフトウェア開発先週プログラム」専任教員である山戸昭三教授、中沢研也教授には、約1年半大変お世話になりました。講義のみならず、進路や就職活動などでも熱心にご指導いただきました。本当にありがとうございました。

また、本プロジェクトで同じチームであった、田中智明君、長野俊輔君、Joao 君にも大変お世話になりました。文書のレビューや技術的なアドバイスをして頂き、ありがとうございました。

最後に、自分の留学を支えてくれた両親や親友に心より感謝致します。

参考文献

- [1] 「SNS 利用者は中毒性とプライバシー侵害のリスクに注意せよ」, マイナビニュース, <http://news.mynavi.jp/news/2011/01/05/076/index.html>.
- [2] 「adidas Japan : お詫び」(アディダスジャパン株式会社), <http://www.adidas.com/jp/corporate/01company/0519.asp>.
- [3] 奥村 学, “ブログマイニング技術の最新動向”, 電子情報通信学会誌, Vol.91, No12, pp.1054-pp.1059, December.2008.
- [4] 奥村 学, “blog マイニング-インターネット上のトレンド, 意見分析を目指して-”, 人工知能学会誌, 21 巻, 4 号, pp.424-429, July.2006.
- [5] 奥村 学, “ブログにおける偏り補正のための書き手のプロファイリング”, 人工知能学会誌, 23 巻, 6 号, pp.798-802, November.2008.
- [6] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel, “A Practical Attack to De-Anonymize Social Network Users”, 2010 IEEE Symposium on Security and Privacy, pp. 223- 238, 2010.
- [7] Home -Twitter 開発者, 参考先 : <https://dev.twitter.com/docs/api>
- [8] Home - Facebook 開発者, <http://developers.facebook.com/>
- [9] Lingua::LanguageGuesser - 言語判定器, 参考先 : http://gensen.dl.itc.u-tokyo.ac.jp/LanguageGuesser/LanguageGuesser_ja.html
- [10] guess-language - 言語判定器, 参考先 : <http://code.google.com/p/guess-language/>
- [11] Google Language API, 参考先 : <http://code.google.com/apis/language/>
- [12] 中谷秀洋, “ナイーブベイズによる言語判定”, 自然言語処理勉強会, 2010.
- [13] WesleyJ Chun, “Core Python Programming, 2nd Edition”, Prentice Hall, pp.654-pp.692, September.2006.
- [14] Randal L. Schwartz, Tom Phoenix & brian d foy, “Learning Perl, Fifth Edition”, O'REILLY, pp.66-pp.104, 2009.
- [15] Marippe_, “色彩センスのいらない配色講座、色は論理的に説明できる”, 参考先 : <http://www.slideshare.net/marippe/ss-9003317>
- [16] 園田誠, “HTML と CSS 基礎から学ぶ Web デザイン”, 秀和システム, pp. 48- 138, September.2010.

- [17]金峰, “DIV+CSS で画面設計”, 人民郵電出版社, pp. 1- 72 , September.2009.
- [18]Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. Automatic text categorization in terms of genre and author. Computational Linguistics, Vol. 26,No. 4, pp. 471–495, 2000.
- [19]池田大輔, 南野朋之, 奥村学, “blog の著者の性別推定”, 言語処理学会全国大会, 2006.
- [20]小早川百合, “話し言葉の男女差-定義・意識・実際”, 日本語ジェンダー学会誌 4 号, 2004.

付録

- 企画段階の成果物
 - ◇ 要件定義書
- 開発段階の成果物
 - ◇ 基本設計書
 - ◇ 詳細設計書
 - ◇ 画面設計書
 - ◇ ER 図
 - ◇ テスト仕様書
- 評価段階の成果物
 - ◇ アンケート
 - ◇ アンケート結果

SNS からの意図しない個人情報漏えい検出システム 要件定義書



背景

ソーシャルネットワーク

現在 Facebook や Twitter に代表されるソーシャルネットワーキングサービス (Social Networking Services : SNS. 以下 SNS) が盛んに利用されている。2011 年 6 月現在、世界規模でサービスを展開する世界最大の SNS である Facebook は約 5 億人のユーザ数を誇る。日本では、mixi や GREE、DeNA などの SNS が多く利用されている。

これら SNS は流行り廃りが激しく、移り変わりが激しいという特徴を持つ。例えば老舗 SNS として一世を風靡した MySpace は現在下火である。

SNS 上では似た趣味や興味を持ったユーザ同士でやり取りが行われることが多い。近年では自分の個人情報をオープンにして利用するユーザや個人情報をオープンに開示するよう求めるサービスも存在する。

SNS におけるプライバシー問題

SNS の利用状況（ユーザ間の相互リンク・つぶやきなど）から個人情報を抽出しようと悪意を持って解析することで、本来匿名であったはずの個人情報が識別されたり、個人を特定するために有用な情報が抽出されてしまったりといった問題（これを脱匿名化の問題という）の懸念が報告されている。昨今では、次のような事例があった。

事例

アディダスジャパンに勤務する社員が、顧客の来店情報を Twitter に書き込み、誹謗中傷を行った。その情報がインターネット上に出回り、社員の過去のツイートを読み、社員の個人情報を特定、その社員は免職になった。

このような事例から、個人情報を直接公開していなくとも、これまでの SNS 利用状況などを解析することで個人情報が特定されうるという脅威を物語っている。

課題

プラグブルなプラットフォームへの要望

SNS 上であるユーザの脱匿名化を行う攻撃手法の研究がいくつも行われている。このような風潮もあり、今後脱匿名化アルゴリズムは解析の精度や速度を向上させたものが次々に現れると予測できる。

さらに、SNS のはやりすたりが激しいことから、脱匿名化の対象となる SNS も変化していくことも予測できる。

このような点を踏まえ、個人推定アルゴリズムの追加や、解析対象となる SNS の追加が容易なプラットフォームが研究者に望まれていると考える。

案その2

上記に述べたように、SNS の現状は拡大状況であり、さまざまなサービスが生まれては消えている。こうした現状もあり、プライバシーに関する研究はまだまだ発展途上である。こうした研究では、新しい攻撃アルゴリズムを試すために、毎回 SNS に応じたデータ収集を行うなどの、と手間が多い。このような問題の解決が望まれている。

目的

我々は、SNS のプライバシー問題に関して、技術的な確立を促進させるために、研究の手助けとなるプラットフォームを作成する。またそのプラットフォーム上で動作するあるユーザの SNS の利用状況を解析し、どのレベルまで個人情報が抽出される可能性があるか、をユーザに通知する Web サービスを作成する。

方針

複数の SNS を組み合わせた解析を行えるプラットフォームを作成する

脱匿名化を行うアルゴリズムの研究が行われ、様々な成果が報告されている。しかし、それらの研究成果をサービスとして SNS ユーザに提供している例は少ない。また存在するサービスのほとんどが、単一の SNS を解析対象としている。そこで、我々は複数の SNS を組み合わせユーザに個人情報の意図しない流出の可能性を提示するサービスを作成する。このように複数の SNS を用いたサービスには新規性があり、有用であると考える。

概要

先に述べた SNS の持つ脱匿名化という課題に対して、我々はユーザの意図しない情報流出の状況を報告するサービス、ならびに情報抽出のためのアルゴリズムを容易に変更しさまざまな手法を検討できるプラットフォームを提供する。

プロジェクトの目標は、ユーザに Web サービスを利用いただき、実際にどの程度ユーザの個人情報が抽出されるかの通知を行うといったデモを目指す。

システム要件

実際のサービス提供において、より正確な情報を通知するためには様々な解析アルゴリズムを開発する必要がある。さらに様々な SNS への対応も必要になる。このような点を考慮し、本プロジェクトでは以下に示す要件を満たすプラットフォームを作成する。

1. 複数の SNS から情報収集を行えること

本プロジェクトで情報収集の対象とする SNS はユーザ数を考慮し、Facebook および Twitter とすること。さらに今後登場する SNS にも対応するため、情報収集機能の追加が可能であること。

2. 複数の SNS を解析対象とすること

1 の要件で取得した SNS 毎の情報の解析、あるいは複数の SNS からの情報を解析可能とすること。

3. Crawler・Analyzer の追加が容易であること

新たな SNS に対するデータの収集・解析を行うプログラムの実装が容易に行えるようデータアクセスや基本的な処理をアーキテクチャレベルでサポートすること、また解析器の実装は Python・Perl・Java など複数の言語で行えること。

4. 解析器の組み合わせが容易に行えること

1 つ以上の解析器の解析結果を利用した解析器の実現が可能であること。

5. 解析結果をユーザに通知できること

ユーザに対して SNS アカウントの情報や解析結果を通知すること。解析結果の通知はテキスト形式だけでなく、グラフなど視覚的な表現も可能とすること。また、作成するシステムにログインしたユーザのみに情報を通知すること。

6. SNS 上での使用言語を特定できること

言語依存の解析アルゴリズムの実装を念頭に置き、SNS 上でユーザが使用している言語(日本語・英語など)を、各情報に対して特定できること。

7. 情報収集、解析のタイミングを制御できること

システムに初めてログインしたユーザに対して優先的に情報収集を行ったり、Crawler や Analyzer に定期的に動作をさせたりといったタスク管理を行うためのスケジューリング機能を有すること。

アーキテクチャ構成

作成するプラットフォームの構成概要を示す。

システム・アーキテクチャ

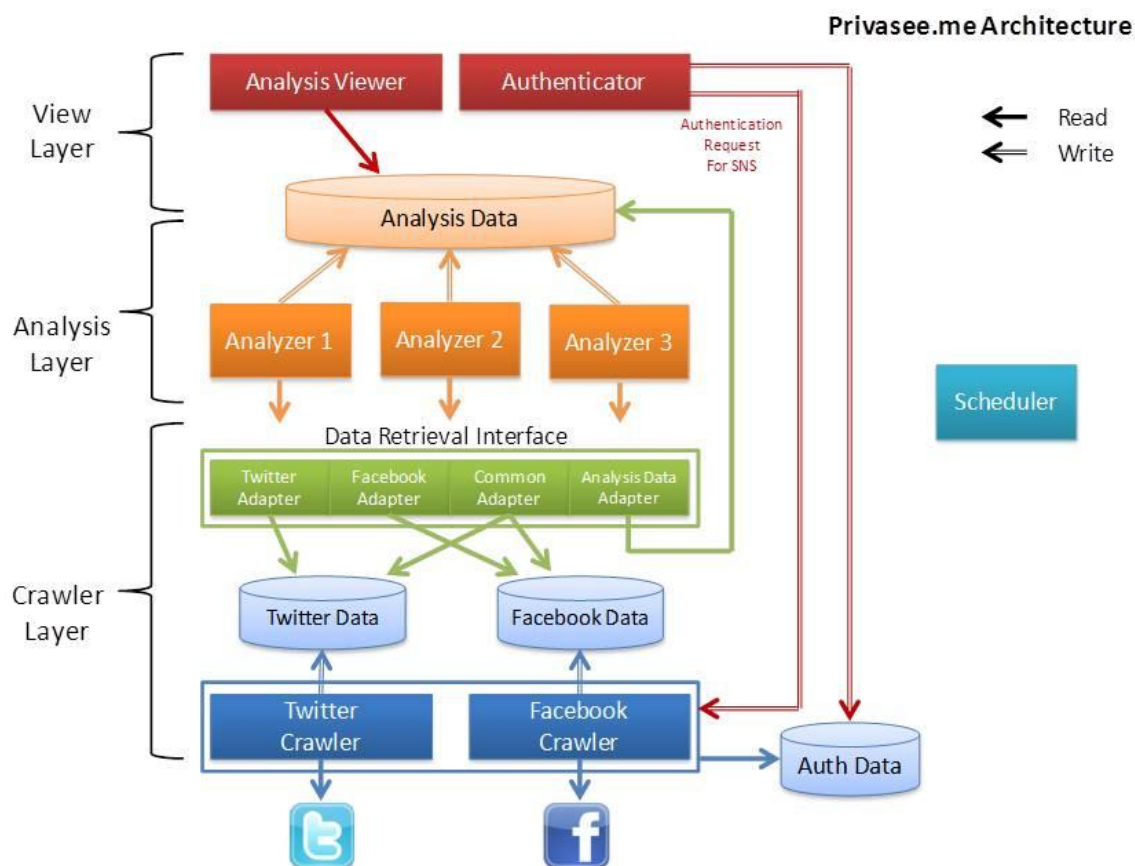


図 1 アーキテクチャの構成

システムの機能

システムの各レイヤーは以下の機能がある。

クローリング・レイヤー

SNS を複数の SNS を利用することができる＝クローラを増やすことができる。

将来的に、Facebook と Twitter 以外の SNS を利用できるよう、拡張性を持たせる。



図 2 複数 SNS を利用できる

ユーザの SNS 上での使用言語の特定機能

サービス利用者の個人情報抽出のために、ユーザの利用言語を特定する必要がある。ユーザが SNS で利用する言語の特定を行う。言語特定は以下の分類問題とする。

日本語・英語・その他

アダプタ・レイヤー

解析機は、クローラを意識しなくてもよい。

解析器とクローラの間にアダプタを挟むことで、解析機は、他のレイヤーの影響を受けない。

解析機・レイヤー

人物属性推定機能

SNS を利用しているユーザの個人情報について、意図せず漏れていると考えられるものをユーザに通知する機能である。通知する項目については、SNS の API を用いて直接取得できる情報・なんらかのアルゴリズムを用いて取得できる情報・通知しない情報の 3 点に分類する。なお、アルゴリズムを用いて解析する情報については、解析できない場合はその旨を通知する。また、アルゴリズムを用いた解析結果の妥当性は保障しない。

表 1 SNS ユーザに通知する個人情報

通知項目	Facebook	Twitter
SNS 上での登録名	○	○
居住地	○	△
所属組織	○	△
性別	○	—

○：API から直接取得 △：解析アルゴリズムを利用し取得 —：未通知

また、SNS の解析アルゴリズムには、日本語文章の解析器であるに Cabocha を用いる。

解析器は1つ以上の SNS のデータを用いることができる。

個人情報の抽出に用いる解析器としては、ある SNS 単独のデータを用いた解析器、または複数の SNS データを用いた解析器の両方を提供できるようにする。



図3 1つ以上の SNS データを解析に用いる

Web インターフェース・レイヤー

ユーザ認証機能

SNS ユーザの個人情報の意図しない流出の可能性を通知するために、SNS からデータを取得する必要がある。データの取得にはユーザの認証が必要となる。このため Web インターフェース・レイヤーでは、ユーザにアプリケーションの利用認証を実現する。

個人情報の通知機能

解析したデータを通知するインターフェースを実現する。

サービスの範囲

対象とする SNS

第1イテレーションでは、Facebook ならびに Twitter を用いた人物属性推定機能を作成する。

運用計画

サービスの運用計画

本サービスは、筑波大学の CS 専攻内での公開を目標とする。

保守計画

運用マニュアル，移行手続き

ソフトウェアの操作マニュアル，操作説明書の作成は開発チームの作成する成果物の範疇とするが，稼動時期の策定や業務への組み込みなどの運用業務，および運用者の変更に備える運用マニュアルの作成については開発チームの行う作業，作成する成果物の範囲外とする。

保守期間

システムおよびソフトウェアの保守作業を平成 23 年 4 月から平成 24 年 3 月末までの間に行うことは努力目標とし，確約はしない。また，平成 24 年 4 月以降は，一切の保証を行わない。

工程計画

以下の開発スケジュールにより開発を行う。

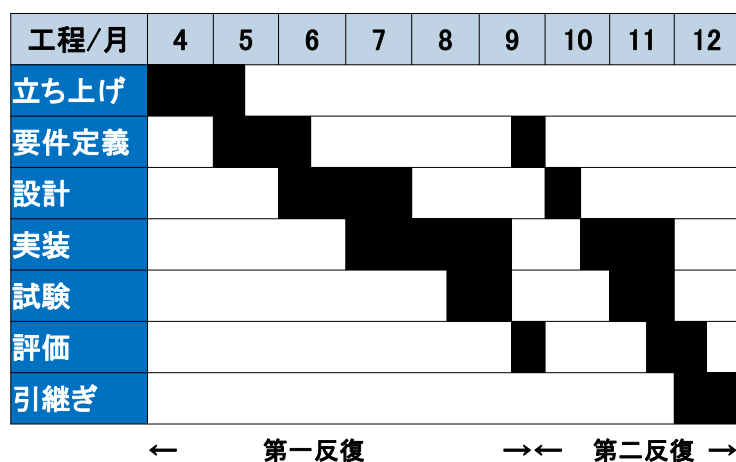


図 4 開発スケジュール

開発体制

開発体制について述べる.

表 2 学生開発チーム体制

氏名	担当
Joao Orui	TBD
田中 智章	TBD
長野 俊輔	TBD
孫 尚君	TBD

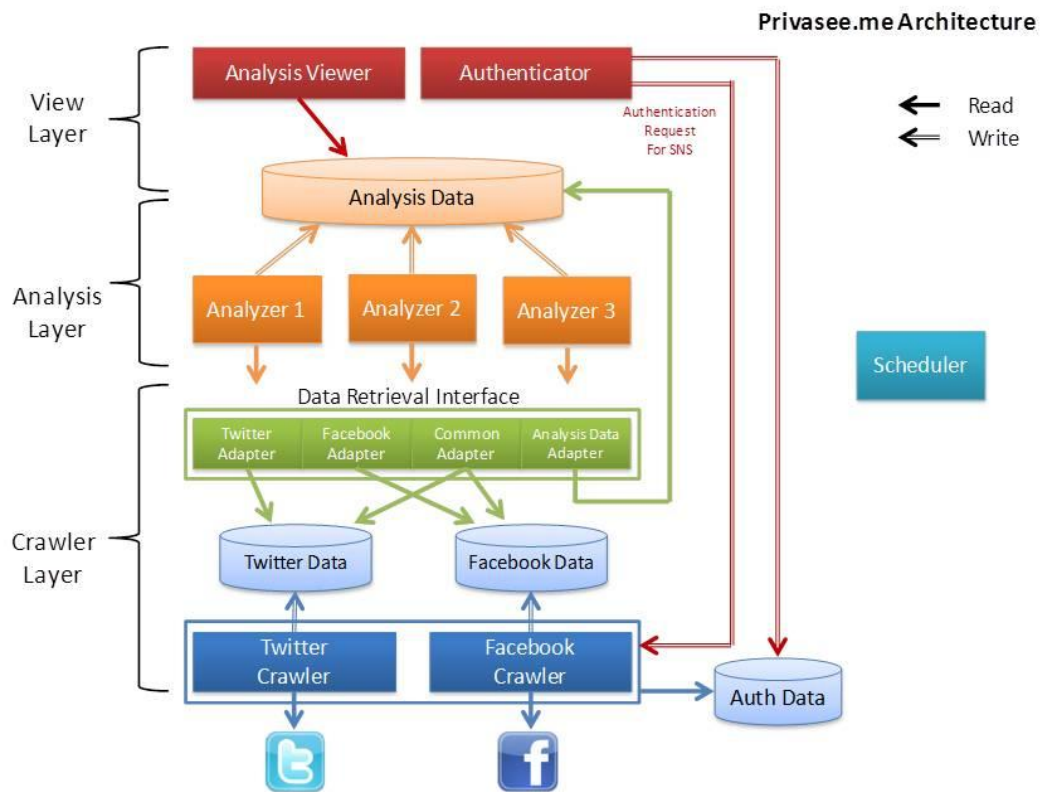
表 3 アドバイザー教員一覧

氏名	担当
佐久間先生	指導教員
山本先生	自然言語処理アドバイザー
中沢先生	プロジェクトマネジメント・アドバイザー
山戸先生	プロジェクトマネジメント・アドバイザー

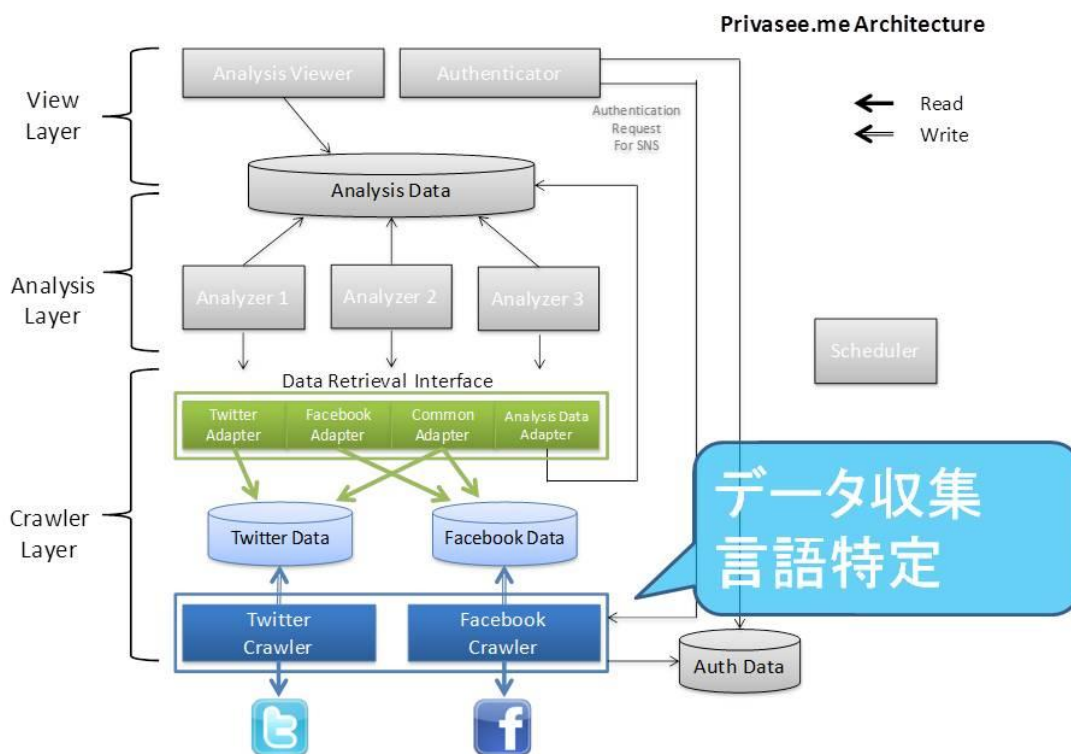
SNS からの意図しない個人情報漏えい検出システム 基本設計書



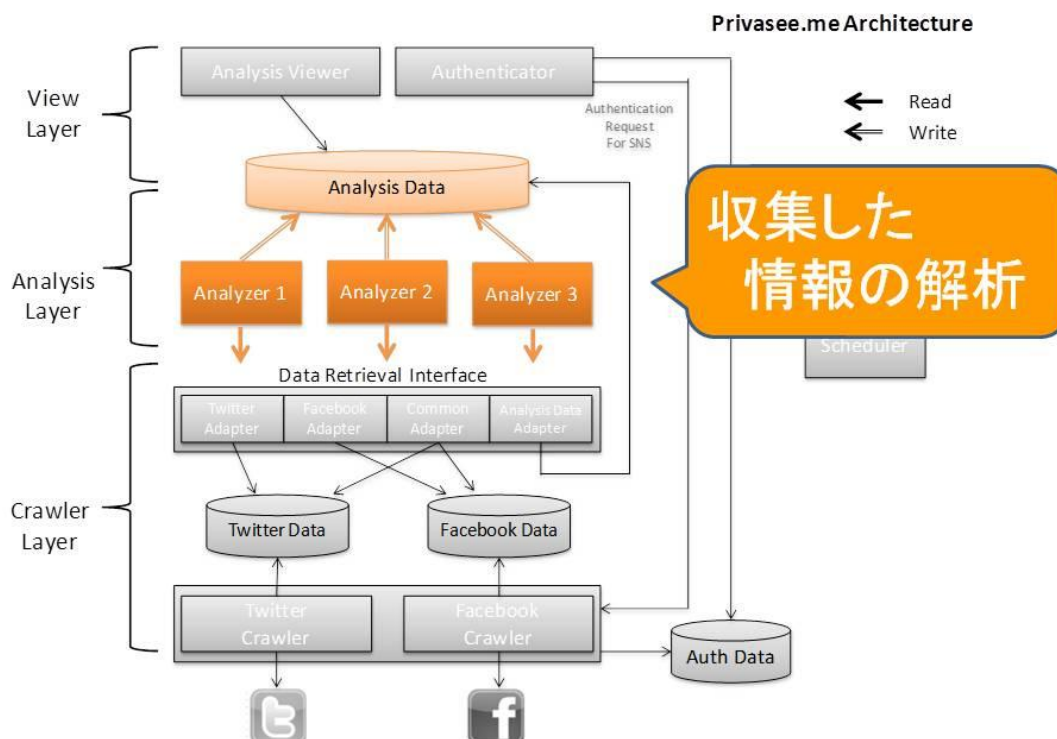
システム・アーキテクチャ



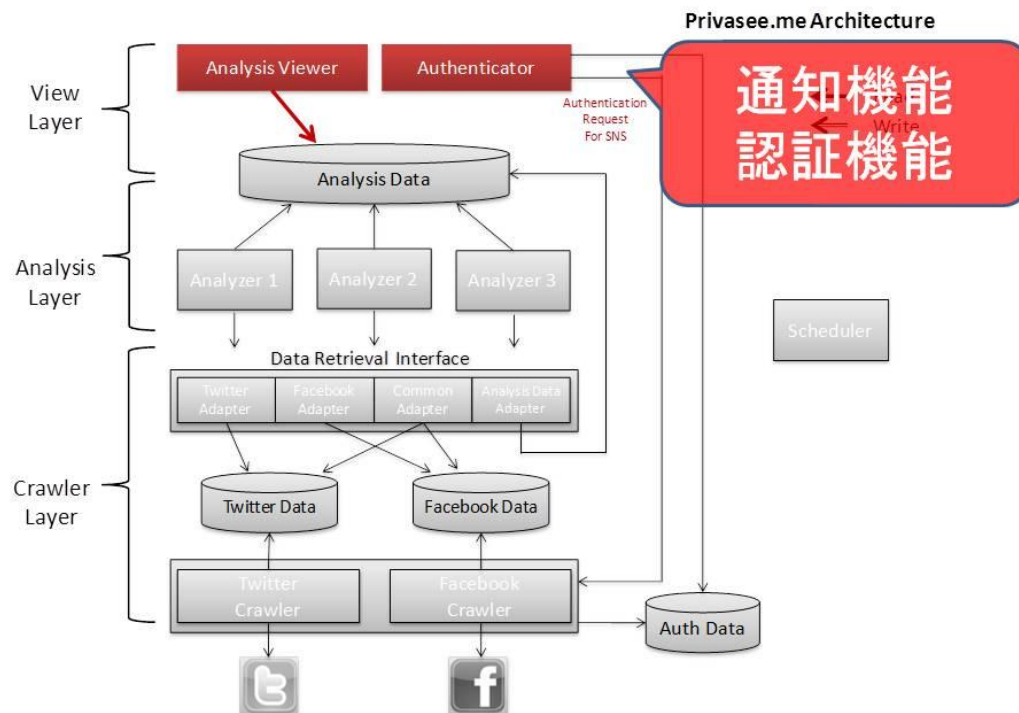
～Crawler Layer～



～Analysis Layer～

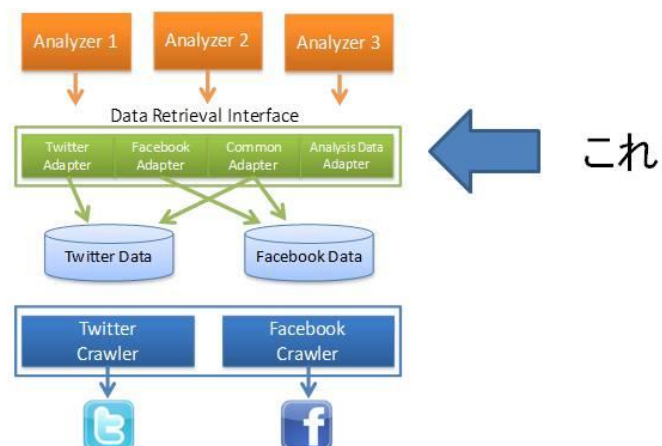


~View Layer~



三層設計

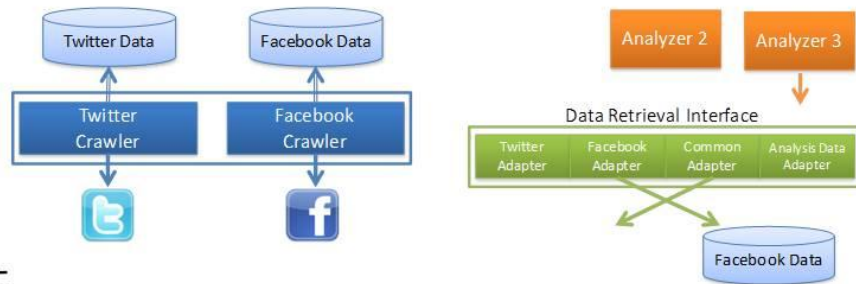
- 3つのレイヤー間の**独立性**を高める
- 解析器・クローラの追加を容易にするためアダプタを設計



Scheduler

要件:「ログイン時に解析を開始する」

想定される問題



すでに

クローラや解析器が動作している場合

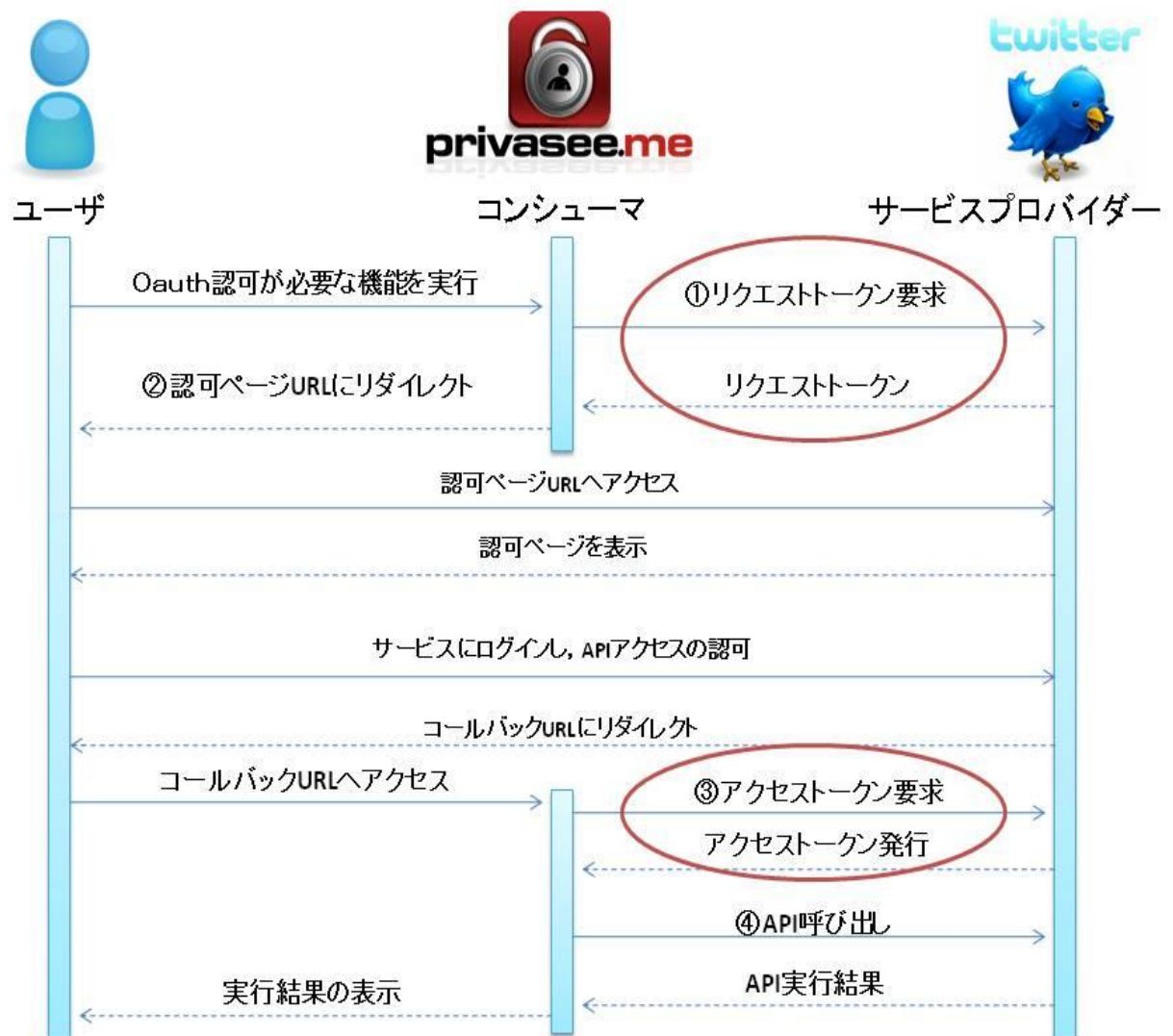
統合的なコンポーネント制御として,
Schedulerを設計に盛り込んだ

認証機能

- 認証の必要性
 - ユーザのセキュリティ保護
 - APIの利用許可
 - APIを利用し, ユーザ情報を収集する
- 認証プロトコル
 - OAuthを利用する...というかこの方法しかない
 - **OAuth1.0**(for Twitter) / OAuth2.0(for Facebook)

OAuthの特徴

- 良い点
 - セキュリティリスクの軽減
 - ユーザ名+パスワード不要
 - APIアクセス範囲をユーザが確認できる
 - 許可の取り消し可
 - オープンな標準規格
- 悪い点
 - 処理がやや複雑(メッセージのやり取りが複数)

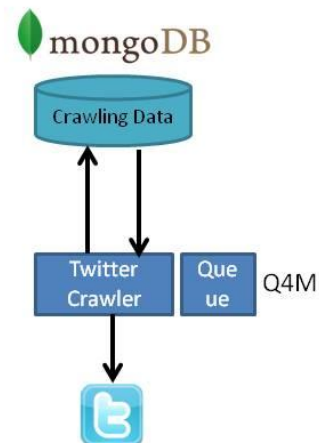


Twitter Crawlerの機能

- ユーザのTL, Profile, Friendsを取得
 - 友人のTLとProfileも取得
- クローリングしたデータを格納
- Crawling Jobを管理する為にQueueを実装
 - Priority Queue
 - 外部からexec()でプログラム呼び出しによるJobの追加

Twitter Crawlerの設計

- Twitterへのアクセス
 - tweepyライブラリを利用
 - 少し改変
- Priority Queueの実装
 - Q4M
 - MySQLのプラグインとして動作
- クローリングしたデータの格納
 - MongoDB





- ドキュメント指向データベース
 - ドキュメント形式で格納
 - keyとvalueのペアからなる構造化された1つのコレクション
 - {key1: value1, key2: value2, key3: value3}
 - 埋め込みドキュメント(ネスト構造)
 - {name: "bob", address: {city: "Tsukuba", prefecture: "Ibaraki"}}
- スキーマレス
 - 事前にドキュメントの構造を定義しなくて良い



- 任意の属性に対してIndexを張れる
- パフォーマンスも良好
 - Indexを張ればMySQL並のパフォーマンス

```
> db.tl.find({"user.screen_name":"jfsso"},"text":1}).sort({"_id":-1}).limit(5)
{"_id" : NumberLong("93646521588924416"), "text" : "お腹空いた！ミーティングまだ終わらない(T_T)" }
{"_id" : NumberLong("93628837379842048"), "text" : "@yum_13 悪くないさ！(笑)" }
{"_id" : NumberLong("93606292400250880"), "text" : "@inohiro 今から向かいます！" }
{"_id" : NumberLong("93512712511946752"), "text" : "サブウェイに寄ってから、大学行く :)" }
{"_id" : NumberLong("93503429732859904"), "text" : "@inohiro 返信した！" }
```

SQL to Mongo Mapping Chart - <http://www.mongodb.org/display/DOCS/SQL+to+Mongo+Mapping+Chart>

MongoDBを採用した理由

- スキーマレス+ドキュメント指向DB
 - Twitter APIで取得したJSONオブジェクトを(ほぼ)そのまま格納できる
 - クローリングしたデータを捨てなくて済む
 - 取得するデータに属性が追加されても、クローラに変更が不要
- ドライバの対応言語, ドキュメントが豊富

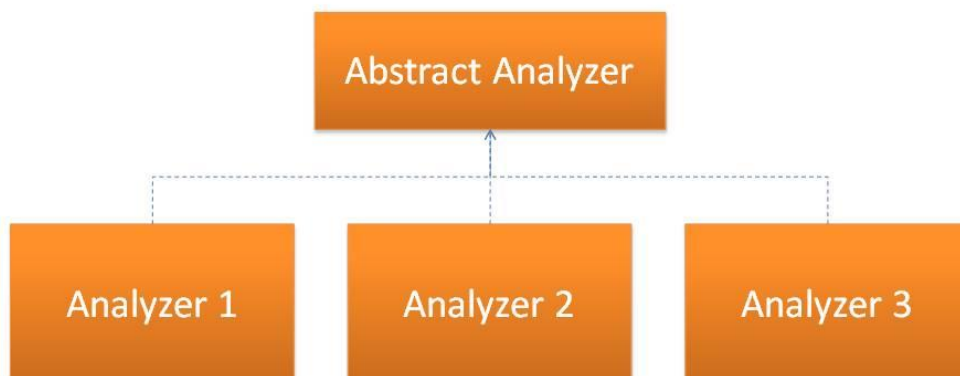
解析機能

- 設計方針

- あとから自由に追加ができる設計
- Interface経由で解析するSNSデータを取得
- 固定な解析項目を出力
 - 名前、生年月日、所属、性別、居住地、など...
- 自由な解析結果も出力可
 - 解析器が項目と表示方法を定義する。

解析器の作成方法

- 共通ロジックを実装した Abstract Analyzerを継承して作成

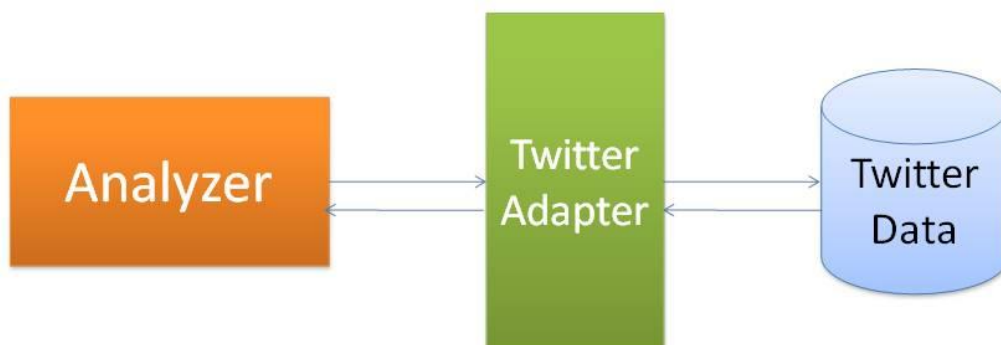


解析器の作成方法

- 各解析器で定義すること
 - 解析に使用するデータ
 - 出力する項目
 - 実行間隔
 - 詳細結果表示可否・表示テンプレート
 - 実行ロジック

データ取得方法

- 各SNS毎にデータ取得用のAdapterを用意
 - 直接データベースを触らず、クローリングが終了したタイミングでデータを適切に取得する
 - 解析器で定義したデータ以外にアクセスを制限



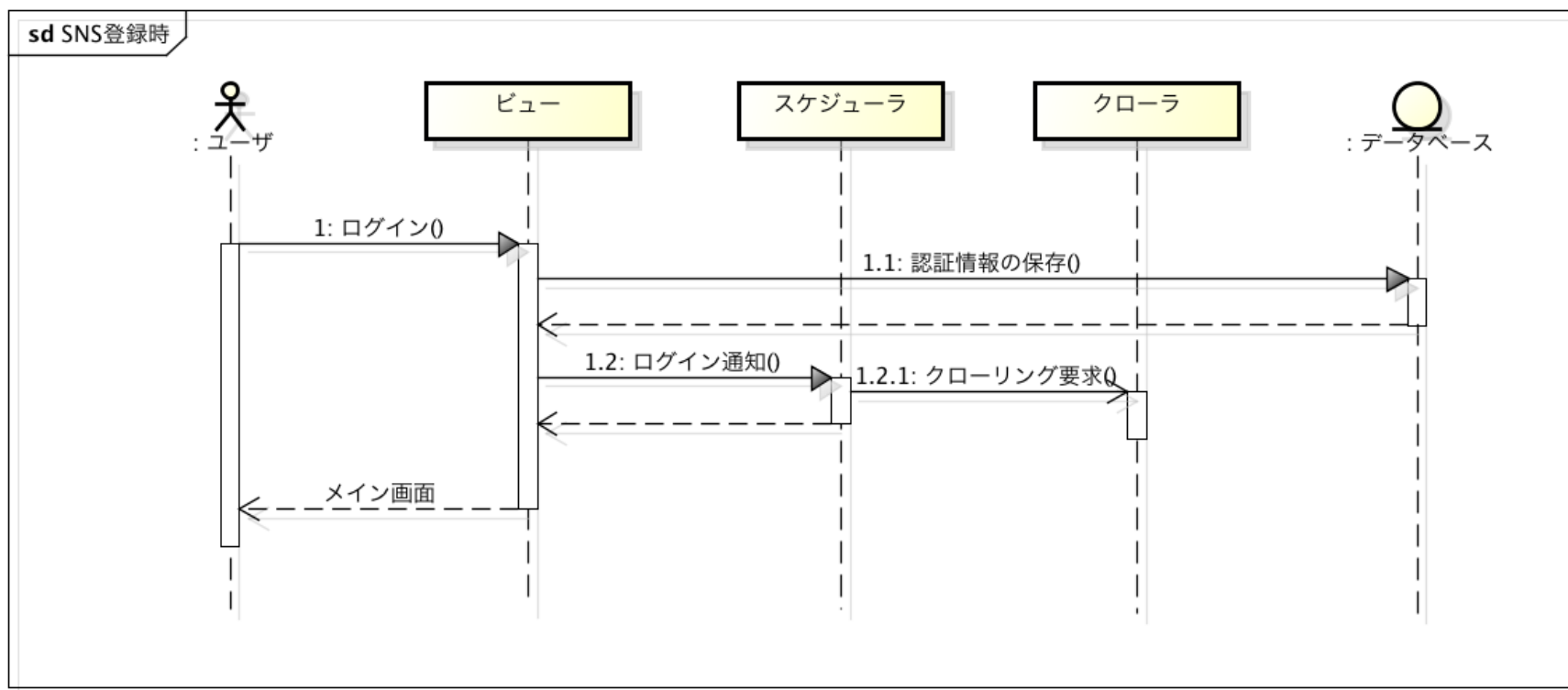
スケジューリング

- 解析器の実行を制御する
 - ※場合によって、
クローラーに対してクローリング要請も行う
- 初回ログイン後にデータがない状態
 - データがないため解析器を実行できない
 - 高優先でクローリングを依頼
 - クローリング終了したら、解析器を実行

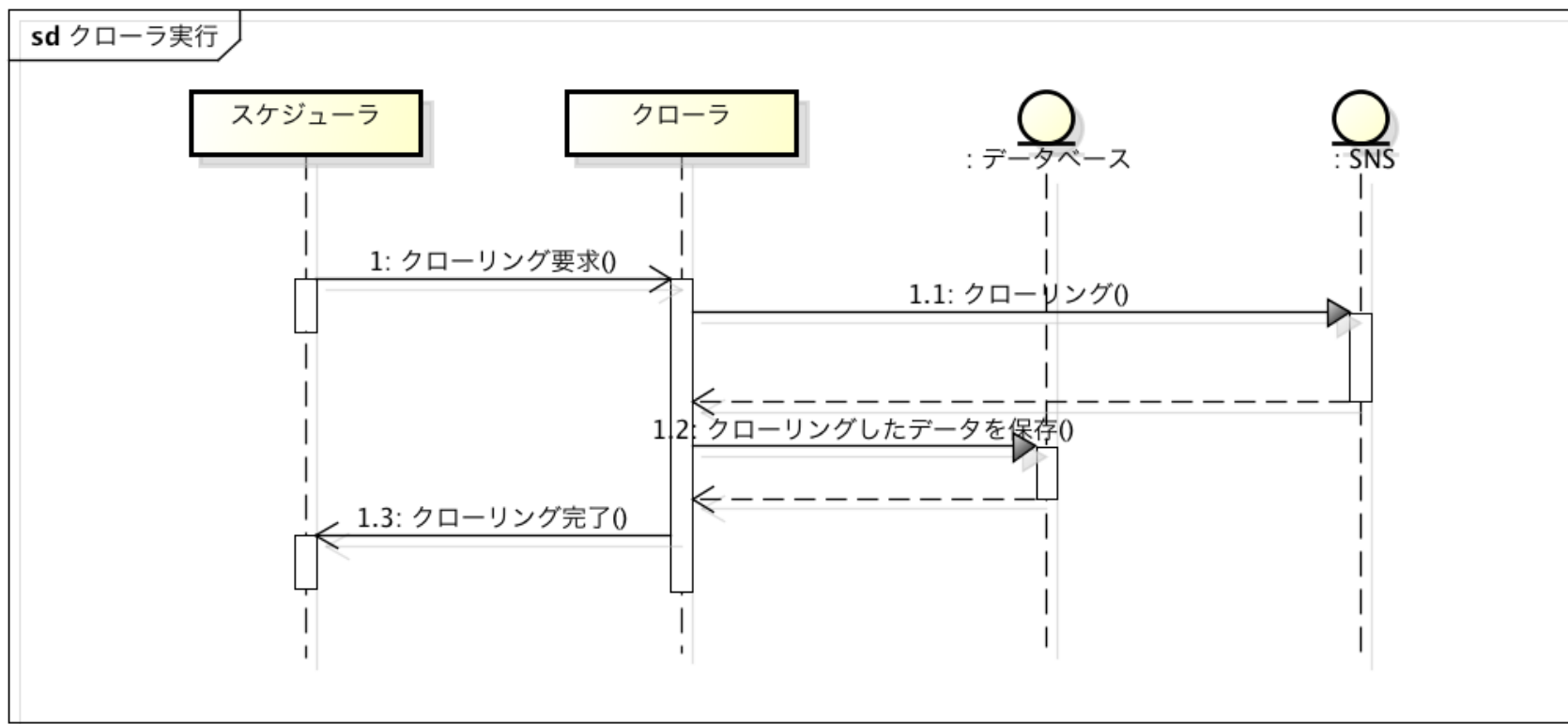
スケジューリング

- Schedulerはサービスとして実行する
- 解析器の実行は必ずSchedulerが判断
 - 定義に依存
 - 実行間隔
 - 利用するデータ

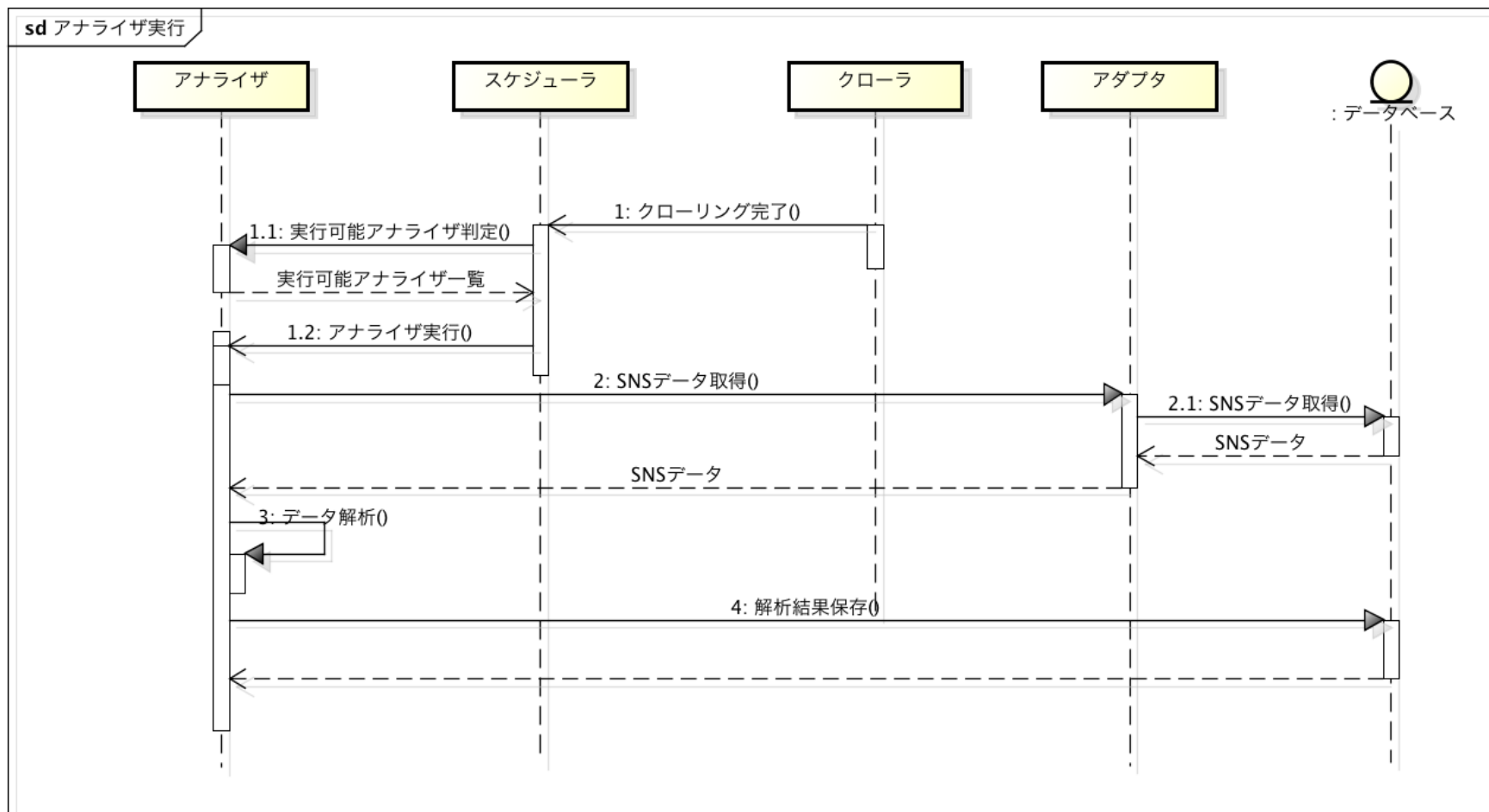
SNS 登録時のシステム動作の流れ



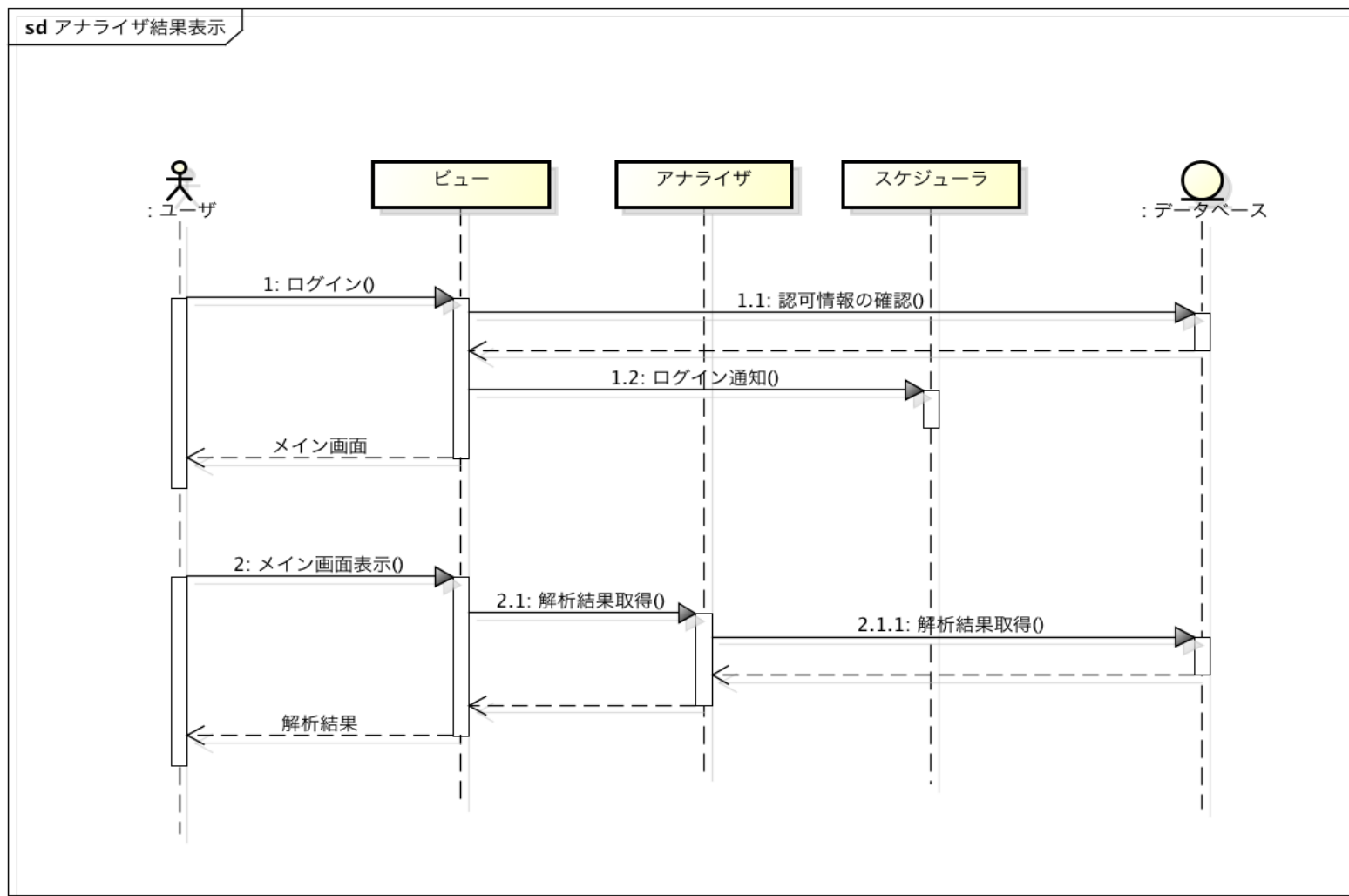
クローラ実行時のシステム動作の流れ



アナライザ実行時のシステム動作の流れ



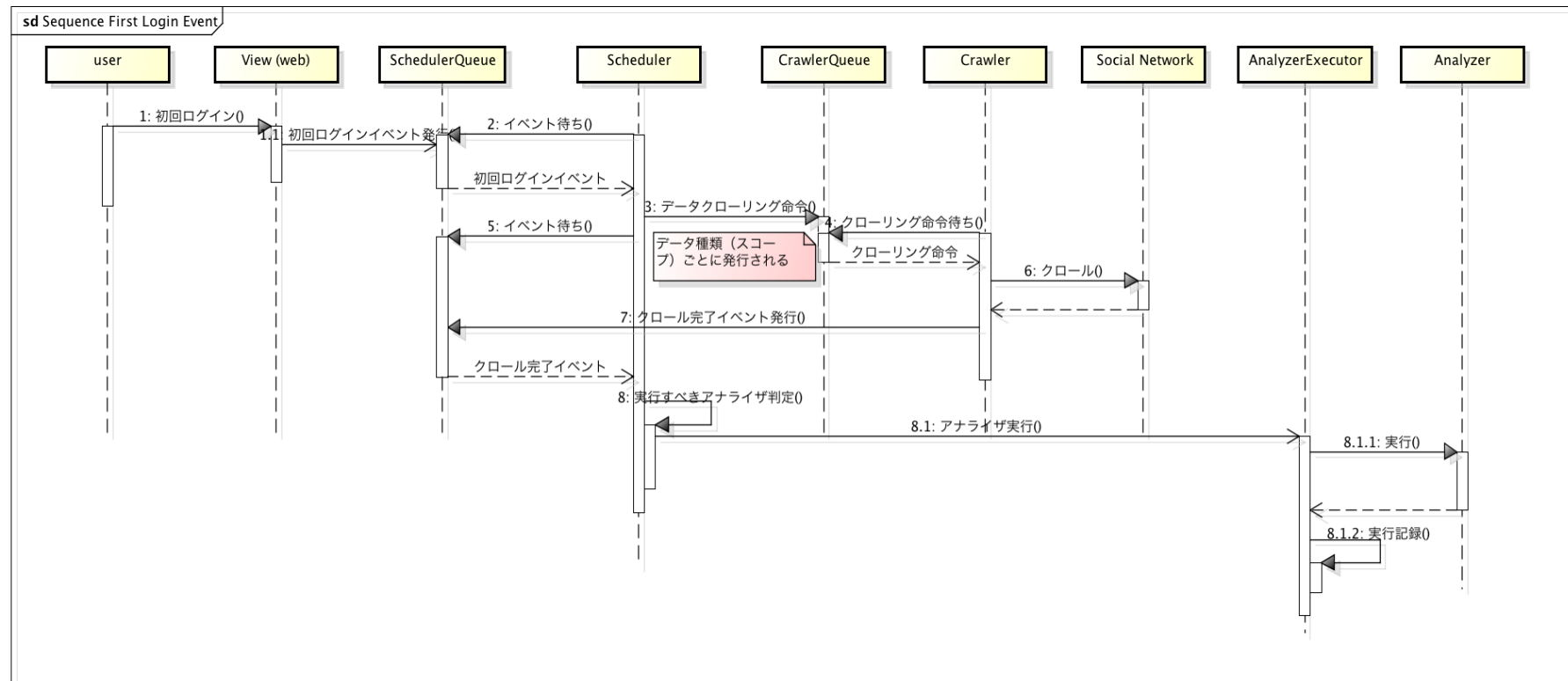
アナライザ結果表示時のシステム動作の流れ



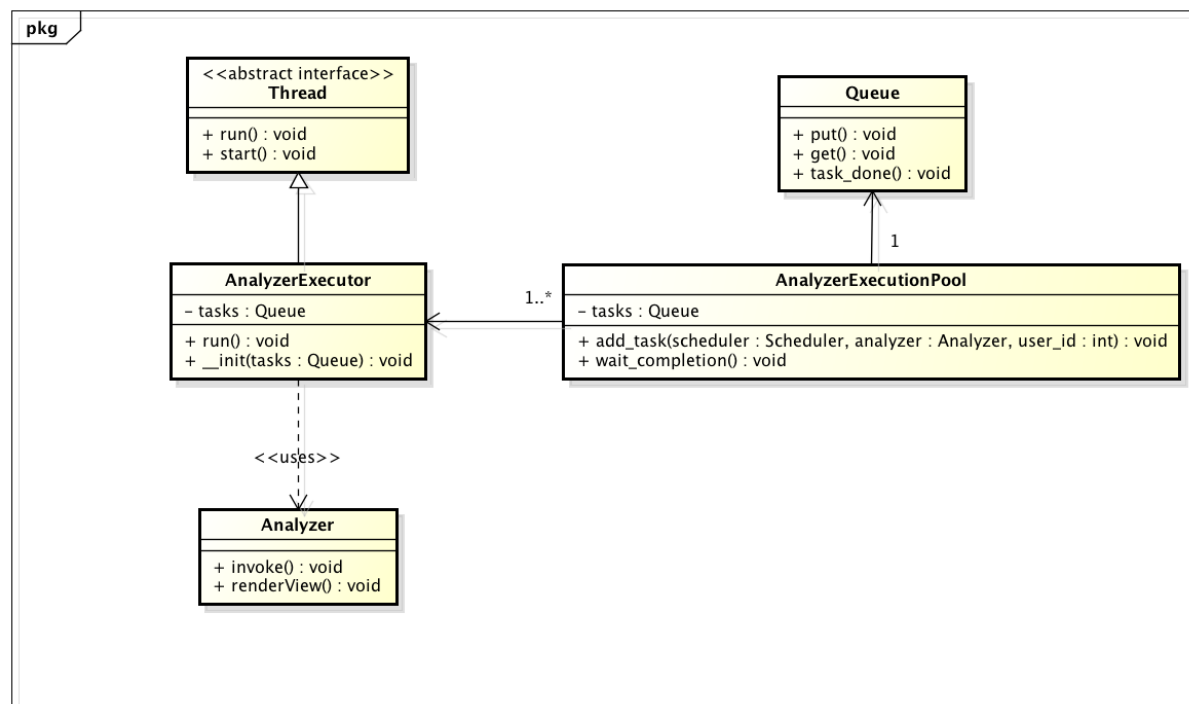
SNS からの意図しない個人情報漏えい検出システム
詳細設計書



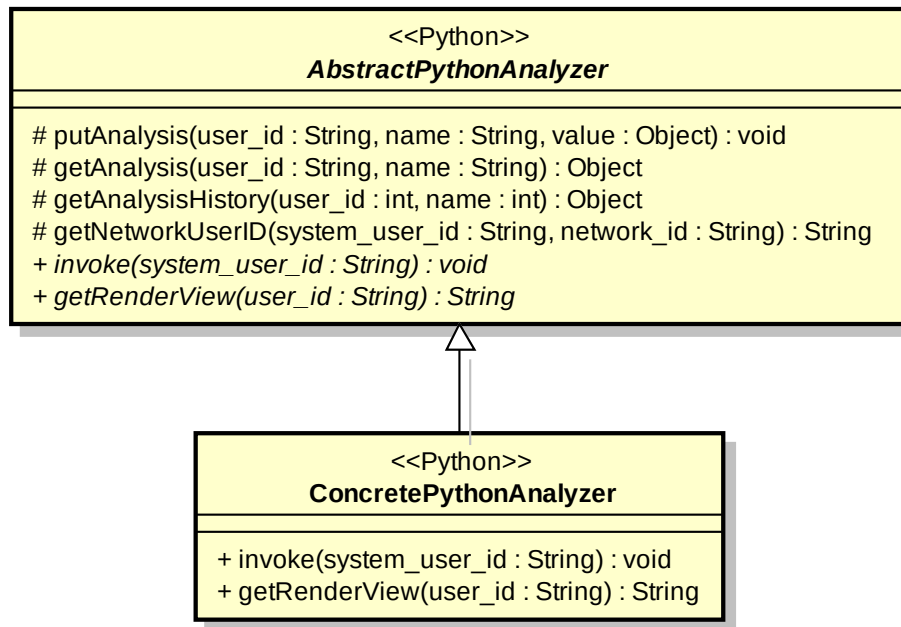
システムログインのシーケンス図



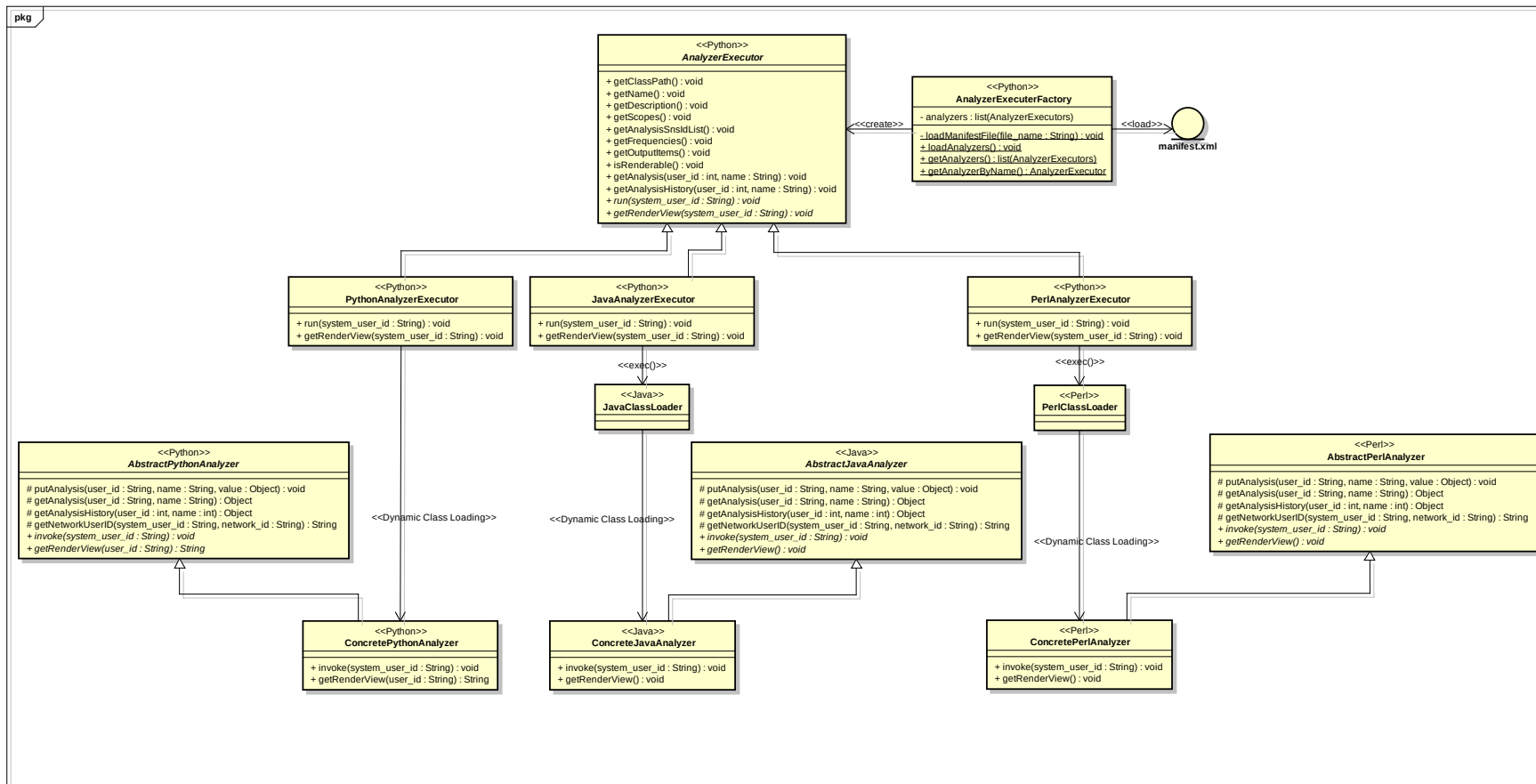
アナライザレイヤのクラス図



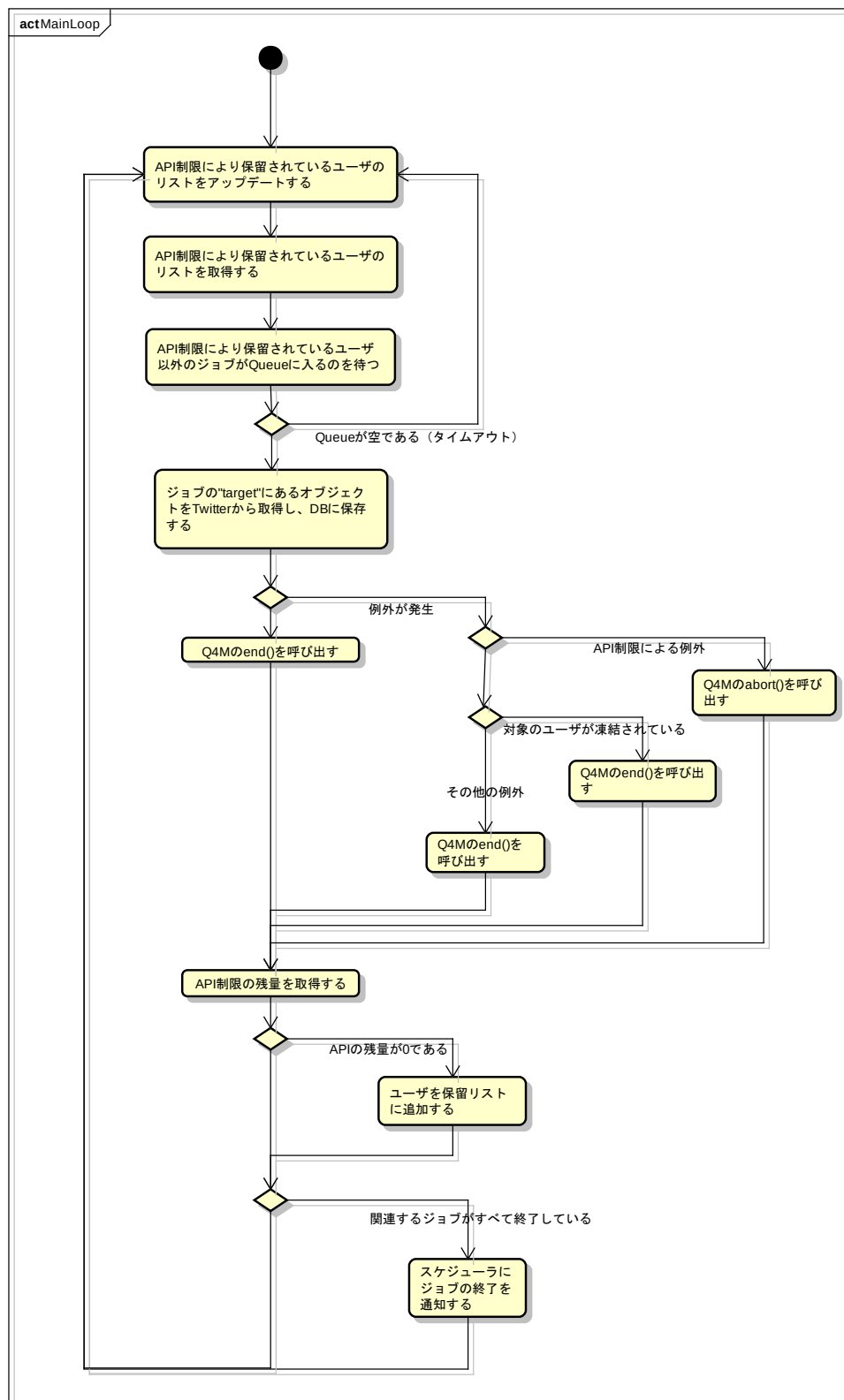
アナライザレイヤのクラス図



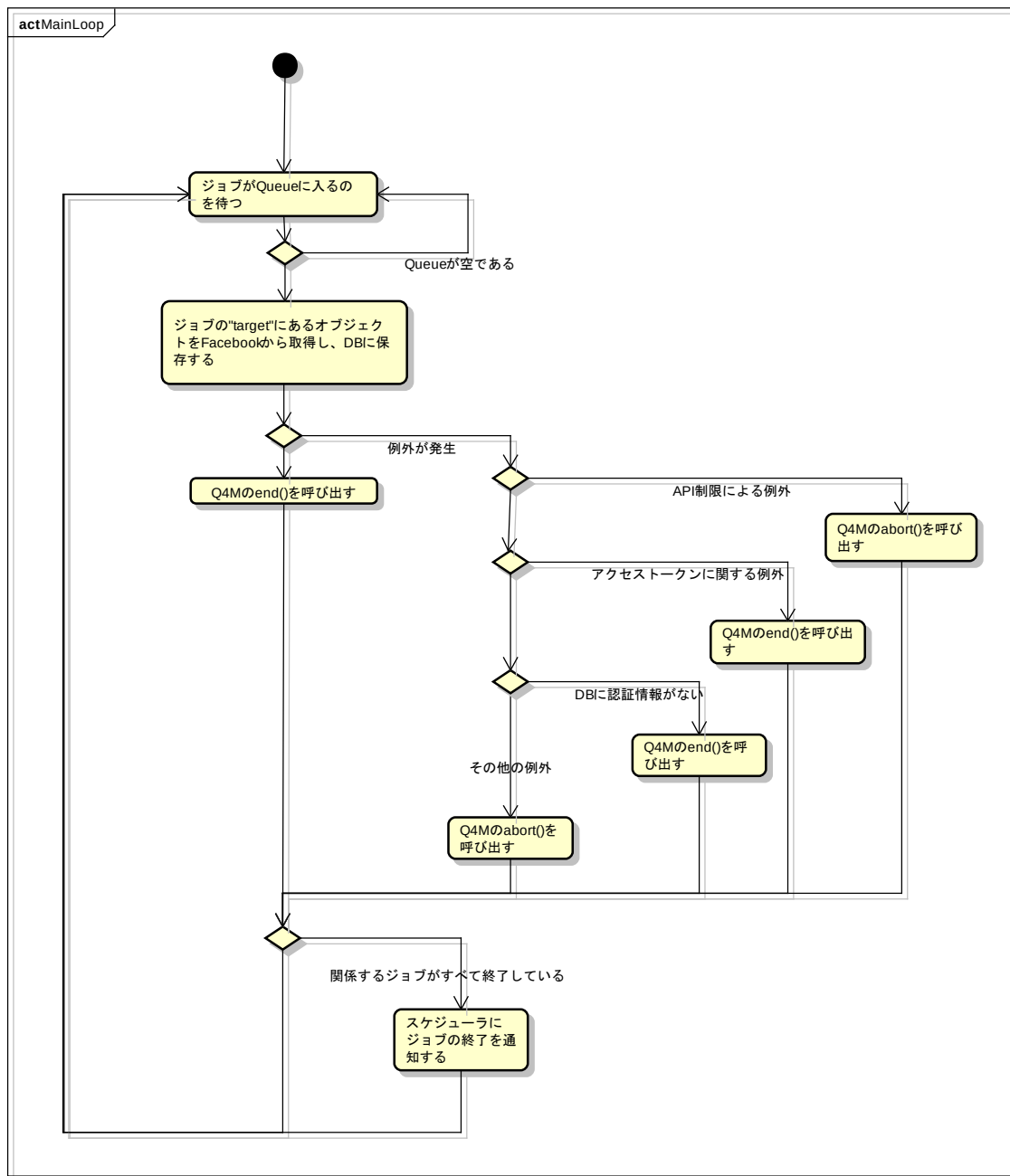
アナライザレイヤのクラス図



Twitter クローラの動作フロー



Facebook クローラの動作フロー



言語判定機能のクラス詳細

1. DetectResult クラスの詳細

このクラスの詳細を以下の表に示す

項目	内容
説明	言語判定の結果を表示するクラスである
パッケージ	crawler.language_detector
クラス名	DetectResult

このクラスのメンバー変数を以下の表に示す

変数名	説明
src	判定するメッセージである
lang_code	言語のコード
language	言語名
confidence	言語判定結果の信頼度(0~1)
is_reliable	信頼できかどうか

1.1. __init__メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	このクラスのコンストラクタである
引数	引数名 : src 型 : String 説明 : 判定の目標となるメッセージである
	引数名 : lang_code 型 : String 説明 : 言語のコード
	引数名 : language 型 : String 説明 : 言語名
	引数名 : confidence 型 : float 説明 : 言語判定結果の信頼度

	引数名 : is_reliable 型 : boolean 説明 : 言語判定結果の信頼度
返り値	なし
アクセス制限	外部からアクセスできる

1.2. get_src メソッド

項目	内容
説明	判定のメッセージを取得するメソッドである
引数	なし
処理	判定のメッセージを返却する
返り値	判定のメッセージ
アクセス制限	外部からアクセスできる

1.3. get_lang_code メソッド

項目	内容
説明	言語のコードを取得するメソッドである
引数	なし
処理	言語のコードを返却する
返り値	言語のコード
アクセス制限	外部からアクセスできる

1.4. get_language メソッド

項目	内容
説明	言語名を取得するメソッドである
引数	なし
処理	言語名のコードを返却する
返り値	言語名のコード
アクセス制限	外部からアクセスできる

1.5. get_confidence メソッド

項目	内容
説明	言語判定結果の信頼度を取得するメソッドである
引数	なし
処理	言語判定結果の信頼度を返却する

返り値	言語判定結果の信頼度
アクセス制限	外部からアクセスできる

1.6. get_is_reliable メソッド

項目	内容
説明	言語判定結果の信頼度を取得するメソッドである
引数	なし
処理	言語判定結果の信頼度を返却する
返り値	言語判定結果の信頼度
アクセス制限	外部からアクセスできる

2. LanguageCodeConverter クラスの詳細

このクラスの詳細を以下の表に示す

項目	内容
説明	言語コードを言語名に変更するクラスである
パッケージ	crawler.language_detector
クラス名	LanguageCodeConverter

このクラスのメンバー変数を以下の表に示す

変数名	説明
languages	以下の表のように、言語コードと言語名のマップを格納する

表：言語コードと言語名のマップ

言語コード：言語名			
'af': 'Afrikaans',	'en': 'English',	'ko': 'Korean',	'sr': 'Serbian',
'sq': 'Albanian',	'eo': 'Esperanto',	'ku': 'Kurdish',	'sd': 'Sindhi',
'am': 'Amharic',	'et': 'Estonian',	'ky': 'Kyrgyz',	'si': 'Sinhalese',
'ar': 'Arabic',	'tl': 'Filipino',	'lo': 'Laothian',	'sk': 'Slovak',
'hy': 'Armenian',	'fi': 'Finnish',	'lv': 'Latvian',	'sl': 'Slovenian',
'az': 'Azerbaijani',	'fr': 'French',	'lt': 'Lithuanian',	'es': 'Spanish',
'eu': 'Basque',	'gl': 'Galician',	'mk': 'Macedonian',	'sw': 'Swahili',
'be': 'Belarusian',	'ka': 'Georgian',	'ms': 'Malay',	'sv': 'Swedish',
'bn': 'Bengali',	'de': 'German',	'ml': 'Malayalam',	'tg': 'Tajik',
'bh': 'Bihari',	'el': 'Greek',	'mt': 'Maltese',	'ta': 'Tamil',
'bg': 'Bulgarian',	'gn': 'Guarani',	'mr': 'Marathi',	'tl': 'Tagalog',
'my': 'Burmese',	'gu': 'Gujarati',	'mn': 'Mongolian',	'te': 'Telugu',
'ca': 'Catalan',	'iw': 'Hebrew',	'ne': 'Nepali',	'th': 'Thai',
'chr': 'Cherokee',	'hi': 'Hindi',	'no': 'Norwegian',	'bo': 'Tibetan',
'zh': 'Chinese',	'hu': 'Hungarian',	'or': 'Oriya',	'tr': 'Turkish',
'zh-CN':	'is': 'Icelandic',	'ps': 'Pashto',	'uk': 'Ukrainian',
'Chinese_simplified',	'id': 'Indonesian',	'fa': 'Persian',	'ur': 'Urdu',
'zh-TW':	'iu': 'Inuktitut',	'pl': 'Polish',	'uz': 'Uzbek',
'Chinese_traditional',	'ga': 'Irish',	'pt-PT':	'ug': 'Uighur',
'hr': 'Croatian',	'it': 'Italian',	'Portuguese',	'vi': 'Vietnamese',
'cs': 'Czech',	'ja': 'Japanese',	'pa': 'Punjabi',	'cy': 'Welsh',
'da': 'Danish',	'kn': 'Kannada',	'ro': 'Romanian',	'yi': 'Yiddish'

'dv': 'Dhivehi', 'nl': 'Dutch',	'kk': 'Kazakh', 'km': 'Khmer',	'ru': 'Russian', 'sa': 'Sanskrit',	
------------------------------------	-----------------------------------	---------------------------------------	--

2.1. getLanguage メソッド

項目	内容
説明	言語コードによって言語名を取得するメソッドである
引数	引数名 : code 型 : String 説明 : 言語コード
処理	languages メンバー変数から code 引数に対応する言語名を取得して、返却する
例外処理	言語コードが存在しない場合 : LanguageDetectionException という例外をスローする、エラーメッセージは”unsupport language for code: 「言語コード」”である
返り値	言語名
アクセス制限	外部からアクセスできる

3. GuessLanguageDetector クラスの詳細

このクラスの詳細を以下の表に示す

項目	内容
説明	Guess Language ライブラリを用いてメッセージに対して言語判定を行うクラスである
パッケージ	crawler.language_detector.guess_language
クラス名	GuessLanguageDetector

3.1. __init__メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	GuessLanguageDetector クラスのコンストラクタである
引数名	なし
処理	「name」メンバー変数に「Guess Language Detector」を設定する。
返回值	なし
アクセス制限	外部からアクセスできる

3.2. detect_message メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	1つのメッセージに対する言語判定を行うメソッドである。
引数名	引数名：msg 型：String 説明：判定の目標となるメッセージである
処理	● Guess Language 言語判定ライブラリの guessLanguageInfo メソッドを呼び出し、msg 引数を渡す。 ● 判定結果により DetectResult クラスのインスタンスを生成し、返す。
返回值	DetectResult クラスのインスタンス
アクセス制限	外部からアクセスできる

3.3. detect_messages メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	複数のメッセージに対する言語判定を行うメソッドである。
引数名	引数名 : <code>msgs</code> 型 : <code>List</code> 説明 : 判定の目標となるメッセージリストである
処理	● リストからメッセージ 1 個ずつ取得し、<code>Guess Language</code> 言語判定ライブラリの <code>guessLanguageInfo</code> メソッドを呼び出し、メッセージを渡す。 ● それぞれの結果をリストにして返却する。
返り値	<code>DetectResult</code> クラスのインスタンスのリスト
アクセス制限	外部からアクセスできる

4. LinguaLanguageDetector クラスの詳細

このクラスの詳細を以下の表に示す

項目	内容
説明	Lingua::LanguageGuess ライブラリを用いてメッセージに対して言語判定を行うクラスである
パッケージ	crawler.language_detector.guess_language
クラス名	LinguaLanguageDetector

4.1. __init__メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	LinguaLanguageDetector クラスのコンストラクタである
引数名	なし
処理	「name」メンバー変数に「Lingua Language Detector」を設定する。
返回值	なし
アクセス制限	外部からアクセスできる

4.2. detect_message メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	1つのメッセージに対する言語判定を行うメソッドである。
引数名	引数名 : msg 型 : String 説明 : 判定の目標となるメッセージである
処理	● Lingua::LanguageGuess 言語判定ライブラリの guessLanguageInfo メソッドを呼び出し、msg 引数を渡す。 ● 判定結果を用いて DetectResult クラスのインスタンスを生成して返す。
返回值	DetectResult クラスのインスタンス
アクセス制限	外部からアクセスできる

4.3. detect_messages メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	複数のメッセージに対する言語判定を行うメソッドである。
引数名	引数名 : msgs 型 : List 説明 : 判定の目標となるメッセージリストである
処理	● リストからメッセージ 1 個ずつ取得し、Lingua::LanguageGuess 言語判定ライブラリのメソッドを呼び出し、メッセージを渡す。 ● それぞれの結果をリストにして返却する。
返り値	DetectResult クラスのインスタンスを含むリスト
アクセス制限	外部からアクセスできる

4.4. detect_message_in_langs メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	指定する言語範囲で複数のメッセージに対する言語判定を行うメソッドである。
引数名	引数名 : msgs 型 : List 説明 : 判定の目標となるメッセージリストである 引数名 : langs 型 : List 説明 : 言語判定の比較範囲
処理	● Lingua::LanguageGuess 言語判定ライブラリの判定範囲を langs 引数に変更する。 ● リストからメッセージ 1 個ずつ取得し、Lingua::LanguageGuess 言語判定ライブラリのメソッドを呼び出し、メッセージを渡す。 ● それぞれの結果をリストにして返却する。
返り値	DetectResult クラスのインスタンスを含むリスト
アクセス制限	外部からアクセスできる

5. GoogleLanguageDetector クラスの詳細

このクラスの詳細を以下の表に示す

項目	内容
説明	GoogleAPI を用いてメッセージに対して言語判定を行うクラスである
パッケージ	crawler.language_detector
クラス名	LanguageDetector

このクラスのメンバー変数を以下の表に示す

変数名	説明
detector_url	使う Google 言語判定の URL アドレスである

5.1. __init__ メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	LanguageDetector クラスのコンストラクタである
引数名	なし
処理	「http://ajax.googleapis.com/ajax/services/language/detect?v=1.0&q=%(message)s」 という文字列を「detector_url」メンバー変数に設定する
返り値	なし
アクセス制限	外部からアクセスできる

5.2. create_detector_url メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	言語判定の URL を生成するメソッドである
引数	引数名 : msg 型 : String 説明 : 判定の目標となるメッセージである
処理	「msg」引数を URL に使える文字列のフォーマットに変更する - 上記で変更した文字列を「detector_url」メンバー変数中の「message」変数に入れ替える

返り値	言語判定を行うリクエスト URL 文字列である
アクセス制限	外部からアクセスできる

5.3. detect_message メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	言語判定の行うメソッドである
引数	引数名 : msg 型 : String 説明 : 言語判定の目標となるメッセージである
処理	● create_detector_url を呼び出して、言語判定用の URL を生成する ● python の urllib と urllib2 モジュールを用いて、言語判定を行う ● json の形でレスポンスから言語判定の結果を解析する ● DetectResult クラスのインスタンスを生成して、言語判定の結果をそれに保存する ● インスタンスを返却する
例外処理	● 言語判定リクエストが失敗した場合 : LanguageDetectionException という例外をスローする、エラーメッセージは “Failed detecting language: 「判定するメッセージ」” である ● 言語判定リクエストがタイムアウトの場合 : LanguageDetectionException という例外をスローする、エラーメッセージは “Failed detecting language (getting 「リクエスト先の URL」 failed): timeout” である ● レスポンスから結果を解析する時に、フォーマット不正などの原因でエラーが発生した場合 : LanguageDetectionException という例外をスローする、エラーメッセージは “Failed detecting language (json failed): 「例外のエラーメッセージ」 ” である ● urllib2.HTTPError, urllib2.URLError が発生した場合 : LanguageDetectionException という例外をスローする、エラーメッセージは “Failed detecting language (getting 「リクエスト先の

	URL」 failed): 「例外の文字列」 "である ● 他の原因で例外が発生した場合 : unknown error LanguageDetectionException という例外をスローする、エラーメッセージは "unknown error" である
返り値	DetectResult クラスのインスタンス
アクセス制限	クラスの中からしかアクセスできない

5.4. detectMessages メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	外部に公開する言語判定機能のメソッドである
引数	引数名 : msgs 型 : String 形のリスト 説明 : 判定の目標となるメッセージリストである
処理	● msgs 引数からそれぞれのメッセージを取得する ● 取得したメッセージに対して、detect_message メソッドを呼び出して、言語判定を行う ● それぞれの結果をリストにして返却する
返り値	DetectResult インスタンスを含むリストである
アクセス制限	外部からアクセスできる

6. CommonLanguageDetector クラスの詳細

このクラスの詳細を以下の表に示す

項目	内容
説明	Lingua::LanguageGuess と Guess Language 二つのライブラリを用いてメッセージに対して言語判定を行うクラスである。
パッケージ	crawler.language_detector
クラス名	CommonLanguageDetector

6.1. __init__メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	CommonLanguageDetector クラスのコンストラクタである
引数名	なし
処理	「Common Language Detector」 という文字列を「name」メンバー変数に設定する
返回值	なし
アクセス制限	外部からアクセスできる

6.2. detect_message メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	言語判定の行うメソッドである
引数	引数名 : msg 型 : String 説明 : 言語判定の目標となるメッセージである
処理	● Guess Language ライブラリを用いてメッセージに対して、言語判定を行う。 ● Guess Language ライブラリが「未知」の結果を返す場合、Guess Language ライブラリを用いてメッセージに対して、もう一度言語判定を行う。 ● DetectResult クラスのインスタンスを生成して、言語判定の結果をそれに保存する。 ● インスタンスを返却する。

返り値	DetectResult クラスのインスタンス
アクセス制限	外部からアクセスできる

6.3. detectMessages メソッド

このメソッドの詳細を以下の表に示す

項目	内容
説明	外部に公開する言語判定機能のメソッドである
引数	引数名： msgs 型： String 形のリスト 説明：判定の目標となるメッセージリストである
処理	● Guess Language ライブラリを用いてメッセージリストに対して、1 回目の言語判定を行う。 ● 判定の結果リストの中で未知という結果があるかどうかを確認する。 ● ある場合、1 回目判定の結果によって新たな言語判定範囲を算出する。 ● Guess Language ライブラリを用いて、1 回目の判定結果が未知のメッセージリストに対して、新たな言語判定範囲で 2 回目の言語判定を行う。 ● DetectResult クラスのインスタンスを生成して、1 回目と 2 回目の判定結果をそれに保存する。 ● インスタンスを返却する。
返り値	DetectResult インスタンスを含むリストである
アクセス制限	外部からアクセスできる

SNS からの意図しない個人情報漏えい検出システム 画面設計書



画面設計書(ユーザ認証ページ)

ユーザ認証ページの画面では下記の図の様に表示される。このページでは、SNSログインボタンが表示される。SNSログインボタンをクリックすることにより解析目標となるSNSが選択される。



ユーザ認証ページのレイアウト詳細



ブラウザ画面



<div>でグループ化した

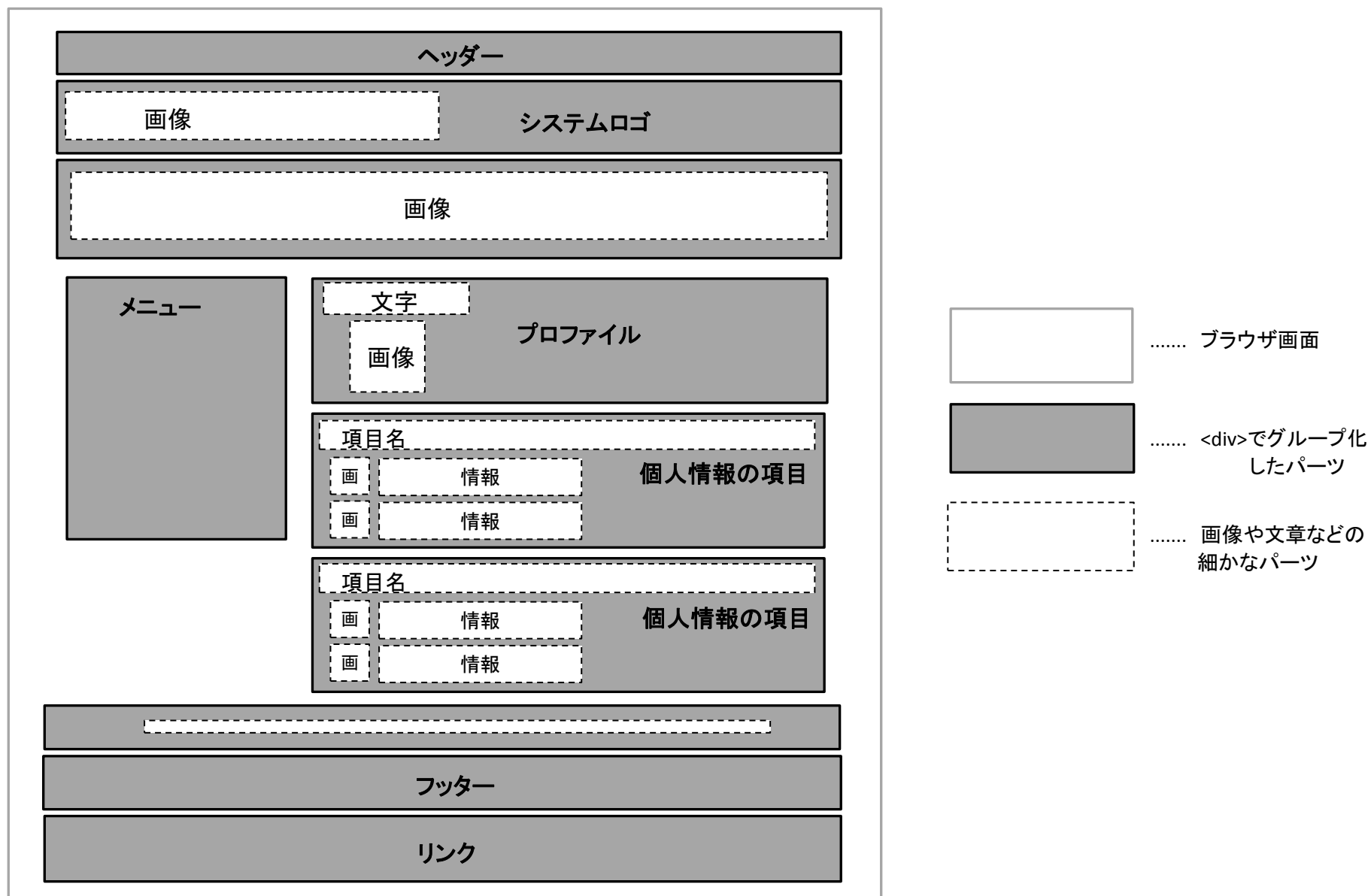


画像や文章などの細かなパーツ

画面設計書(解析結果表示ページ)



ユーザ認証ページのレイアウト詳細











SNS からの意図しない個人情報漏えい検出システム データベース ER 図



SNS からの意図しない個人情報漏えい検出システム テスト仕様書



単体試験 - 言語判定モジュール

モジュール名	説明	作成日	作成チーム	担当者
言語判定	ユーザの発言に対して言語判定を行うモジュールである	2011/10/1	TeamPrivasee	孫

試験番号	クラス名	メソッド名	試験・確認項目	試験・確認内容	可否	期日	担当
ut1001	LanguageCodeConverter	getLanguage	languagesインスタンス変数からcode引数に対応する言語名を取得して、返却する	getLanguageメソッドを呼び出し、“en”文字列をパラメータとして渡す。返り値が“English”であることを確認する	合格	2011/10/1	孫
ut1002	LanguageCodeConverter	getLanguage	languagesインスタンス変数からcode引数に対応する言語名を取得して、返却する	getLanguageメソッドを呼び出し、“ja”文字列をパラメータとして渡す。返り値が“Japanese”であることを確認する	合格	2011/10/1	孫
ut1003	LanguageCodeConverter	getLanguage	languagesインスタンス変数からcode引数に対応する言語名を取得して、返却する	getLanguageメソッドを呼び出し、“zh-CN”文字列をパラメータとして渡す。返り値が“Chinese_simplified”であることを確認する	合格	2011/10/1	孫
ut1004	LanguageCodeConverter	getLanguage	言語コードが存在しない場合：LanguageDetectionExceptionという例外をスローする、エラーメッセージは“unsupport language for code:「言語コード」”である	getLanguageメソッドを呼び出し、“noexist”文字列をパラメータとして渡す。 getLanguageメソッドからLanguageDetectionExceptionをcatchして、エラーメッセージを確認する	合格	2011/10/1	孫
ut1005	DetectResult	get_src	DetectResultのインスタンスを生成して、src変数に“this is a test”文字列を設定する	get_srcメソッドを呼び出して、“this is a test”文字列を返却する	合格	2011/10/1	孫
ut1006	DetectResult	get_lang_code	DetectResultのインスタンスを生成して、lang_code変数に“en”文字列を設定する	get_lang_codeメソッドを呼び出して、“en”文字列を返却する	合格	2011/10/1	孫
ut1007	DetectResult	get_lang	DetectResultのインスタンスを生成して、language変数に“english”文字列を設定する	get_langメソッドを呼び出して、“language”文字列を返却する	合格	2011/10/1	孫
ut1008	DetectResult	get_confidence	DetectResultのインスタンスを生成して、confidence変数に0.7777を設定する	get_confidenceメソッドを呼び出して、0.7777を返却する	合格	2011/10/1	孫
ut1009	DetectResult	get_is_reliable	DetectResultのインスタンスを生成して、is_reliable変数にTrueを設定する	get_is_reliableメソッドを呼び出して、0.7777文字列を返却する	合格	2011/10/1	孫

ut1010	GoogleLanguageDetector	create_detector_url	引数を「detector_url」インスタンス変数 中での「message」変数に入れ替えるこ と	create_detector_urlメソッドを呼び出して、 “thisisatest”文字列を渡して、戻り値を確認する	合格	2011/10/1	孫
ut1011	GoogleLanguageDetector	create_detector_url	「msg」引数をURLに使える文字列の フォーマットに変更する	create_detector_urlメソッドを呼び出して、“this is a test”文字列を渡す。戻り値の中で、 “this+is+a+test”という文字列を確認する	合格	2011/10/1	孫
ut1012	GoogleLanguageDetector	detectMessage	英語メッセージを判定でき、結果を正し く返却できること	“this is a test for language detection”という文字 列を渡して、判定の結果を確認する	合格	2011/10/1	孫
ut1013	GoogleLanguageDetector	detectMessage	日本語メッセージを判定でき、結果を正 しく返却できること	“Pythonを使ってプログラミングの学習を開始され る方を対象として、Pythonを使ったプログラムの記 述方法や実行までをサンプルを使いながら順に学 習していきます。” という文字列を渡して、判定の結果を確認する	合格	2011/10/1	孫
ut1014	GoogleLanguageDetector	detectMessage	中国語メッセージを判定でき、結果を正 しく返却できること	“周日一天根本恢复不过来, 中午饭还没吃就已 经不行了吃完估计要猝死了”という文字列を渡し て、判定の結果を確認する	合格	2011/10/1	孫
ut1015	GoogleLanguageDetector	detectMessage	言語判定リクエストが失敗した場合: LanguageDetectionExceptionという例外 をスローする、エラーメッセージは “Failed detecting language:「判定する メッセージ」”である	間違ったURLにアクセスして、 LanguageDetectionExceptionをcatchして、エラー メッセージを確認する	合格	2011/10/1	孫
ut1016	GoogleLanguageDetector	detectMessages	複数のメッセージを判定でき、結果を正 しく返却できること	“this is a test for language detection”、 “Pythonを使ってプログラミングの学習を開始され る方を対象として、Pythonを使ったプログラムの記 述方法や実行までをサンプルを使いながら順に学 習していきます。”、 “周日一天根本恢复不过来, 中午饭还没吃就已 经不行了吃完估计要猝死了” という複数の文字列を渡して、判定結果を確認す る	合格	2011/10/1	孫

ut1017	GuessLanguageDetector	detectMessage	メッセージを判定でき、結果を正しく返却できること	“Pythonを使ってプログラミングの学習を開始される方を対象として、Pythonを使ったプログラムの記述方法や実行までをサンプルを使いながら順に学習していきます。”という文字列を渡して、判定の結果を確認する	合格	2011/10/1	孫
ut1018	GuessLanguageDetector	detectMessage	メッセージを判定できない場合は、「UNKNOWN」を返却すること	detectMessageメソッドを呼び出して、“My name is jack”という短い文字列を渡して、判定の結果が「UNKNOWN」であることを確認する	合格	2011/10/1	孫
ut1019	GuessLanguageDetector	detectMessages	複数のメッセージを判定でき、結果を正しく返却できること	“this is a test for language detection”、“Pythonを使ってプログラミングの学習を開始される方を対象として、Pythonを使ったプログラムの記述方法や実行までをサンプルを使いながら順に学習していきます。”、“周日一天根本恢复不过来，中午饭还没吃就已经不行了吃完估计要猝死了”という複数の文字列を渡して、判定結果を確認する	合格	2011/10/1	孫
ut1020	LinguaLanguageDetector	detectMessage	メッセージを判定でき、結果を正しく返却できること	“Pythonを使ってプログラミングの学習を開始される方を対象として、Pythonを使ったプログラムの記述方法や実行までをサンプルを使いながら順に学習していきます。”という文字列を渡して、判定の結果を確認する	合格	2011/10/1	孫
ut1021	LinguaLanguageDetector	detect_message_in_langs	指定している言語範囲で複数のメッセージに対する言語判定を行うこと	「Pour comprendre votre monde devient ennuyeuse à mes yeux ..」というフランス語、「Y ahora, una pausa para la publicidad..」というスペイン語、「从此以后我要把时间花在更多有意义的事情上。」という中国語を含むメッセージリストと、“en”、“jp”という言語範囲リストを渡す。判定の結果が“英語”、“英語”、“日本語”であることを確認する。	合格	2011/10/1	孫

ut1022	LinguaLanguageDetector	detectMessages	複数のメッセージを判定でき、結果を正しく返却できること	“this is a test for language detection”、 “Pythonを使ってプログラミングの学習を開始される方を対象として、Pythonを使ったプログラムの記述方法や実行までをサンプルを使いながら順に学習していきます。”、 “周日一天根本恢复不过来, 中午饭还没吃就已经不行了吃完估计要猝死了” という複数の文字列を渡して、判定結果を確認する	合格	2011/10/1	孫
ut1023	CommonLanguageDetector	detectMessage	与えられたメッセージに対して、まず、Guess Languageライブラリを用いて言語判定を行うこと	“Pythonを使ってプログラミングの学習を開始される方を対象として、Pythonを使ったプログラムの記述方法や実行までをサンプルを使いながら順に学習していきます。” “という長い文字列を渡す。判定の結果を通じて、判定を行ったクラスはGuessLanguageDetectorクラスであることを確認する	合格	2011/10/1	孫
ut1024	CommonLanguageDetector	detectMessage	Guess Languageライブラリが「未知」の結果を返す場合、Lingua::LanguageGuessライブラリを用いて、もう一度言語判定を行う。	detectMessageメソッドを呼び出して、“My name is jack”という短い文字列を渡す。判定の結果を通じて、判定を行ったクラスはLinguaLanguageDetectorクラスであることを確認する	合格	2011/10/1	孫
ut1025	CommonLanguageDetector	detectMessages	Guess Languageライブラリが「未知」の結果を返す場合、Lingua::LanguageGuessライブラリを用いて、もう一度言語判定を行う。	detectMessageメソッドを呼び出して、“My name is jack”という短い文字列を渡す。判定の結果を通じて、判定を行ったクラスはLinguaLanguageDetectorクラスであることを確認する	合格	2011/10/1	孫

ut1026	CommonLanguageDetector	detectMessages	与えられたメッセージリストに対して、Guess Languageライブラリを用いて1回目の言語判定を行う。1回目判定の結果によって新たな言語判定範囲を算出する。Guess Languageライブラリを用いて、1回目の判定結果が未知のメッセージリストに対して、新たな言語判定範囲で2回目の言語判定を行う。	「Pour comprendre votre monde devient ennuyeuse à mes yeux ..」というフランス語、「Y ahora, una pausa para la publicidad..」というスペイン語、「Music is the best cure for all of your problems. 」という英語、「My name is jack 」という英語を含むメッセージリストを渡す。 以下の2点を確認する: ①判定の結果が“フランス語”、“スペイン語”、“英語”、“英語”であること ②判定を行ったクラスが “GuessLanguageDetector”、 “GuessLanguageDetector”、 “GuessLanguageDetector”、 “LinguaLanguageDetector”であること	合格	2011/10/1	孫

単体試験書(ユーザ認証ページ)

試験番号	試験・確認項目	試験方法	合否	期日	担当
ut2001	詳細設計書(ユーザ認証ページ)と同様のレイアウトの画面が表示される。	レイアウト、ラベルを目視確認	合格	2011/9/17	孫
ut2002	Facebookログインボタン	クリックで、Facebookのログイン画面に遷移することを確認	合格	2011/9/17	孫
ut2003	Twitterログインボタン	クリックで、Twitterのログイン画面に遷移することを確認	合格	2011/9/17	孫
ut2004	システムロゴの表示	システムロゴが画面の真ん中に表示されることを確認	合格	2011/9/17	孫
ut2005	システム目的情報の表示	「あなたの「プライバシー」を守りましょう」という文字が表示されることを確認	合格	2011/9/17	孫
ut2006	システム説明情報の表示	「あなたが使っているSNSが他人からどのようにみられているか解析します」という文字が表示されることを確認	合格	2011/9/17	孫
ut2007	区切り線の表示	区切り線が詳細設計書(ユーザ認証ページ)と同様に表示されることを確認	合格	2011/9/17	孫
ut2008	認証説明の表示	「まずは利用しているSNSからPrivasee.Meにログインしましょう」という文字が表示されることを確認	合格	2011/9/17	孫
ut2009	著作権説明の表示	「Copyright(C) 2011-2012 Team Privasee, All Rights Reserved」という文字が表示されることを確認	合格	2011/9/17	孫

単体試験書(解析結果表示ページ)

試験番号	試験・確認項目	試験方法	合否	期日	担当
ut2101	詳細設計書(ユーザ認証ページ)と同様のレイアウトの画面が表示される	レイアウト、ラベルを目視確認	合格	2011/9/18	孫
ut2102	トップ画像の表示	システムロゴが詳細設計書(解析結果表示ページ)と同様に表示されることを確認	合格	2011/9/18	孫
ut2103	Facebookログインボタン	クリックで、Facebookのログイン画面に遷移することを確認	合格	2011/9/18	孫
ut2104	Twitterログインボタン	クリックで、Twitterのログイン画面に遷移することを確認	合格	2011/9/18	孫
ut2105	ユーザ写真の表示	ユーザ写真が表示できることを確認	合格	2011/9/18	孫
ut2106	個人情報の項目は数により自動に延長・短縮ができる	目視確認	合格	2011/9/18	孫
ut2107	個人情報の項目名の表示	個人情報の項目名が詳細設計書(解析結果表示ページ)と同様に表示されることを確認	合格	2011/9/18	孫
ut2108	解析された情報の表示	解析された情報が詳細設計書(解析結果表示ページ)と同様に表示されることを確認	合格	2011/9/18	孫
ut2109	メニューの項目数はアナライザ数より変更できる	アナライザ数を変更することで、メニューでの項目数を確認	合格	2011/9/18	孫
ut2110	メニューでのアナライザのリンク	クリックで、該当するアナライザ解析結果表示画面に遷移することを確認	合格	2011/9/18	孫
ut2111	個人情報を解析するアナライザ	「Copyright(C) 2011-2012 Team Privasee, All Rights Reserved」という文字が表示されることを確認	合格	2011/9/18	孫

SNS からの意図しない個人情報漏えい検出システム

アンケート



対象評価項目	なし
質問 1 アンケートへの同意について質問いたします。 今回の調査結果を、氏名など本人が特定できないよう、匿名化した上で、統計データなどの形式で研究目的で利用/公表されることに同意していただけるでしょうか。 できる限り、ご協力をお願いいたします。	
()	はい
()	いいえ

対象評価項目	利用の状況																				
質問 2 どのような方が privasee.me を利用して頂いたか、についての質問です 質問 2-1 所属学類あるいは専攻を選んでください。 選択肢（以下 17 項目から選択） <table border="1"> <tr> <td>人文・文化学群</td> <td>社会・国際学群</td> <td>人間学群</td> <td>生命環境学群</td> </tr> <tr> <td>理工学群</td> <td>情報学群</td> <td>医学部</td> <td>体育・芸術専門学群</td> </tr> <tr> <td>教育研究科</td> <td>人文社会科学 研究科</td> <td>ビジネス科学 研究科</td> <td>数理物質科学 研究科</td> </tr> <tr> <td>システム情報 工学研究科</td> <td>生命環境科学 研究科</td> <td>人間総合科学 研究科</td> <td>図書館情報 メディア研究科</td> </tr> <tr> <td>その他</td> <td colspan="3"></td> </tr> </table>		人文・文化学群	社会・国際学群	人間学群	生命環境学群	理工学群	情報学群	医学部	体育・芸術専門学群	教育研究科	人文社会科学 研究科	ビジネス科学 研究科	数理物質科学 研究科	システム情報 工学研究科	生命環境科学 研究科	人間総合科学 研究科	図書館情報 メディア研究科	その他			
人文・文化学群	社会・国際学群	人間学群	生命環境学群																		
理工学群	情報学群	医学部	体育・芸術専門学群																		
教育研究科	人文社会科学 研究科	ビジネス科学 研究科	数理物質科学 研究科																		
システム情報 工学研究科	生命環境科学 研究科	人間総合科学 研究科	図書館情報 メディア研究科																		
その他																					
質問 2-2 性別を選択してください。																					
()	男性																				
()	女性																				
質問 2-3 privasee.me に登録した SNS を選択してください。 登録した SNS とは、ログインに利用した SNS・ログイン後新たに追加した SNS を意味します																					
()	Twitter で利用した																				
()	Facebook で利用した																				
()	Twitter と Facebook の両方で利用した																				
()	利用していない																				

対象評価項目	利用の状況																																										
質問 3 普段利用している SNS の種類や、その用途についてお聞きします。																																											
質問 3-1 よく利用する SNS を選択してください。（複数回答可）																																											
☐	Twitter																																										
☐	Facebook																																										
☐	mixi																																										
☐	Google+																																										
☐	人人(れんれん)																																										
☐	その他																																										
質問 3-2 SNS の利用目的や利用方法をそれぞれ選択してください。																																											
利用していない SNS に関しては、「利用していない」を選択してください。																																											
	<table border="1"> <thead> <tr> <th></th><th>利用していない</th><th>アカウントはあるが、ほぼ利用していない</th><th>自分の投稿がメイン</th><th>友人・周囲の投稿を見るのがメイン</th><th>自分でよく投稿し、友人の投稿もよく見る</th></tr> </thead> <tbody> <tr> <td>Twitter</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td></tr> <tr> <td>Facebook</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td></tr> <tr> <td>mixi</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td></tr> <tr> <td>Google+</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td></tr> <tr> <td>人人(れんれん)</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td></tr> <tr> <td>その他</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td></tr> </tbody> </table>		利用していない	アカウントはあるが、ほぼ利用していない	自分の投稿がメイン	友人・周囲の投稿を見るのがメイン	自分でよく投稿し、友人の投稿もよく見る	Twitter	☐	☐	☐	☐		Facebook	☐	☐	☐	☐		mixi	☐	☐	☐	☐		Google+	☐	☐	☐	☐		人人(れんれん)	☐	☐	☐	☐		その他	☐	☐	☐	☐	
	利用していない	アカウントはあるが、ほぼ利用していない	自分の投稿がメイン	友人・周囲の投稿を見るのがメイン	自分でよく投稿し、友人の投稿もよく見る																																						
Twitter	☐	☐	☐	☐																																							
Facebook	☐	☐	☐	☐																																							
mixi	☐	☐	☐	☐																																							
Google+	☐	☐	☐	☐																																							
人人(れんれん)	☐	☐	☐	☐																																							
その他	☐	☐	☐	☐																																							
質問 3-3 SNS では見ず知らずのユーザが、あなたの投稿などを閲覧することが可能な場合があります。あなたは、第三者に見られていることを意識して SNS を利用していますか？																																											
例えば、Twitter では、あなたのことをフォローしていないユーザからもあなたの投稿を閲覧することが可能な場合があります。																																											
☐	第三者から閲覧される可能性があることを知らなかった																																										
☐	可能性は知っているが、特に意識していない																																										
☐	可能性は知っており、そのことを承知であえて情報をオープンにしている																																										
☐	第三者に見られる可能性を考慮して、公開する情報と公開しない情報の使い分けをしている																																										
☐	その他																																										

対象評価項目	利用の状況
質問 3-4 多くの SNS では、自分のプロフィールや投稿などの情報の公開範囲を、「プライバシー設定」を利用することで制限できます。あなたは設定を理解し、適切に公開範囲を定めていますか？	
()	設定の存在を知らなかった
()	設定の存在は知っているが、よくわからないので適当に設定している
()	設定を理解しているが、特に変更の必要を感じず、デフォルトのままである
()	設定を理解したうえで、適切に使い分けをしていると思う
()	その他
質問 3-5 privasee.me のような SNS のアプリケーション連携を行う際、どの情報にアクセスできるか（パーミッション）を要求します。あなたはパーミッションを確認して、アプリケーション連携の認証を行っていますか？ SNS のアプリ連携を行う際は必ず、アプリケーションがアカウントのどのデータにアクセスすることができるかを表すパーミッションの一覧が表示されています。	
()	パーミッションの存在を知らなかった
()	パーミッションについては知っているが、よくわからないので適当に認証している
()	パーミッションを理解しているが、心配ないと考えており、すぐに認証している
()	パーミッションを読み、その内容を理解したうえで安全を確認し、認証を行っている（パーミッションによっては、連携をキャンセルすることもある）
()	その他
質問 3-6 SNS を利用しているうえで、知らない人からの友人申請が来た場合について、あてはまるものを選択してください。（複数回答可） SNS によって使い方は異なると思いますが、あなたのよく利用する SNS に置き換え回答してください。	
()	友人申請はほとんど来ない
()	見ず知らずの人で匿名のアカウントの場合は、ほぼ認証を行わない
()	見ず知らずの人で匿名のアカウントの場合は、認証を行わないことが多い
()	見ず知らずの人で実名のアカウントの場合は、認証することが多い
()	見ず知らずの人で実名のアカウントの場合は、ほとんど認証する

対象評価項目	ユーザインターフェース (UI)				
質問 4 privasee.me の使用感・操作感・全体のデザインに関して質問いたします。					
質問 4-1 privasee.me へのログインは簡単に行えましたか？					
	1	2	3	4	
難しかった	()	()	()	()	簡単だった
質問 4-2 privasee.me のデザインは見やすいものでしたか？					
	1	2	3	4	
見にくい	()	()	()	()	見やすい
質問 4-3 privasee.me に配置されているアイコンやテキストなどは、簡単に理解できましたか？					
	1	2	3	4	
理解しにくかった	()	()	()	()	理解しやすかった
質問 4-4 全体的に評価して、privasee.me の操作感はどうでしたか？					
	1	2	3	4	
わかりにくい. 操作しにくい	()	()	()	()	わかりやすい. 操作しやすい

対象評価項目	主観的満足度				
質問 5					
質問 5-1 privasee.me へのログイン時間はどうか？					
	1	2	3	4	
気になります (遅いと感じる)	()	()	()	()	特に気になりません (早いと感じる)
質問 5-2 ログイン後，解析結果が表示されるまでの時間は気にならないものでしたか？					
	1	2	3	4	
気になります (遅いと感じる)	()	()	()	()	特に気になりません (早いと感じる)

対象評価項目	ユーザインタフェース
質問 6 privasee.me のデザインや GUI に関して，良かった点あるいは悪かった点など，気がついたことがありましたら，自由にお書きください.	

対象評価項目	解析の正確性と効果																																																																																
質問 7 privasee.me が解析した個人情報についての質問です。 注意 privasee.me はあなたの SNS アカウントが「周囲の人（第三者）からどのように見られているか」という視点で個人情報を解析しています。 質問 7-1 privasee.me が解析したあなたの個人情報について、それぞれ該当するものを選んでください。それぞれの解析結果について、もっともあてはまるものを選んでください。 <table border="1"> <thead> <tr> <th></th><th>結果が表示されなかった</th><th>一致したもの・しないものが混じっていた</th><th>だいたい一致した</th><th>完全に一致した</th></tr> </thead> <tbody> <tr> <td>名前</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>性別</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>誕生日</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>所属</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>居住地</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>出身地</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> </tbody> </table> 質問 7-2 個人情報の解析結果、あるいは解析方法の説明を読み、あなたはどのように感じましたか。それぞれの解析結果について、もっともあてはまるものを選んでください。 <table border="1"> <thead> <tr> <th></th><th>結果や説明が表示されず、何ともいえない</th><th>結果が違っていたので、何ともいえない</th><th>参考になる部分はあるが、特に情報漏えいの意識は変わらない</th><th>解析方法を読み、情報漏えいに対する意識が高まった</th><th>解析結果を見て、個人情報漏えいに気をつけようと思った</th></tr> </thead> <tbody> <tr> <td>名前</td><td>()</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>性別</td><td>()</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>誕生日</td><td>()</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>所属</td><td>()</td><td>()</td><td>()</td><td>()</td><td>()</td></tr> <tr> <td>居住地</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>出身地</td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>						結果が表示されなかった	一致したもの・しないものが混じっていた	だいたい一致した	完全に一致した	名前	()	()	()	()	性別	()	()	()	()	誕生日	()	()	()	()	所属	()	()	()	()	居住地	()	()	()	()	出身地	()	()	()	()		結果や説明が表示されず、何ともいえない	結果が違っていたので、何ともいえない	参考になる部分はあるが、特に情報漏えいの意識は変わらない	解析方法を読み、情報漏えいに対する意識が高まった	解析結果を見て、個人情報漏えいに気をつけようと思った	名前	()	()	()	()	()	性別	()	()	()	()	()	誕生日	()	()	()	()	()	所属	()	()	()	()	()	居住地						出身地					
	結果が表示されなかった	一致したもの・しないものが混じっていた	だいたい一致した	完全に一致した																																																																													
名前	()	()	()	()																																																																													
性別	()	()	()	()																																																																													
誕生日	()	()	()	()																																																																													
所属	()	()	()	()																																																																													
居住地	()	()	()	()																																																																													
出身地	()	()	()	()																																																																													
	結果や説明が表示されず、何ともいえない	結果が違っていたので、何ともいえない	参考になる部分はあるが、特に情報漏えいの意識は変わらない	解析方法を読み、情報漏えいに対する意識が高まった	解析結果を見て、個人情報漏えいに気をつけようと思った																																																																												
名前	()	()	()	()	()																																																																												
性別	()	()	()	()	()																																																																												
誕生日	()	()	()	()	()																																																																												
所属	()	()	()	()	()																																																																												
居住地																																																																																	
出身地																																																																																	

対象評価項目	解析の正確性と効果				
質問 8 privasee.me によるその他の解析についての質問です。 質問 8-1 privasee.me のサイドバーに表示される解析情報が何を示しているか、意図を理解することができましたか？もっともあてはまるものをそれぞれ選択してください。					
	結果が表示されなかった	結果が何を表しているか、全くわからなかった	結果の意図を理解できた		
ベストフレンド解析	()	()	()		
頻出表現解析	()	()	()		
性別判定	()	()	()		
居住地推定解析器	()	()	()		
質問 8-2 privasee.me のサイドバーに表示される解析器の解析結果は、自分の個人情報と一致していましたか？					
	結果が表示されなかった	一致したもの・しないものが混じっていた	だいたい一致した	完全に一致した	
ベストフレンド解析	()	()	()	()	
頻出表現解析	()	()	()	()	
性別判定	()	()	()	()	
居住地推定解析器	()	()	()	()	
質問 8-3 privasee.me のサイドバーに表示される解析器の説明文や解析結果を、どのように感じましたか？					
	結果や説明が表示されず、何ともいえない	結果が違っていただけ、何ともいえない	参考になる部分はあるが、特に情報漏えいの意識は変わらない	解析方法を読み、情報漏えいに対する意識が高まった	解析結果を見て、個人情報漏えいに気をつけようと思った
ベストフレンド解析	()	()	()	()	()
頻出表現解析	()	()	()	()	()
性別判定	()	()	()	()	()
居住地推定解析器	()	()	()	()	()

対象評価項目	解析の正確性と効果
質問 9 解析結果全体についての質問です.	
質問 9-1 解析結果全体の正答率はどうでしたか？あてはまるもの一つを選らんで下さい	
()	正答率は低いと感じた
()	どちらかと言えば正答率は低いと感じた
()	どちらかと言えば正答率は高いと感じた
()	正答率は高いと感じた
()	その他

対象評価項目	解析の正確性と効果
質問 10 privasee.me の解析結果に関して、良かった点あるいは悪かった点など、気がついたことがありましたら、自由にお書きください.	

対象評価項目	主観的満足度				
質問 11 privasee.me を利用していただき感じた満足感について					
質問 11-1 privasee.me 全体を通して、満足できるサービスだと感じましたか？					
	1	2	3	4	
満足できない	()	()	()	()	満足できる
質問 11-2 SNS からの個人情報漏えいを知るため、privasee.me を繰り返し利用したいと思えますか？ コンセプト等から判断してください。					
	1	2	3	4	
使いたいとは思わない	()	()	()	()	使いたいと思う
質問 11-3 どのような改良点があると、今後も繰り返し privasee.me を利用したいと思いますか？ (複数回答可)					
()	どうあっても、利用しない				
()	解析対象 SNS の種類が増えれば利用すると思う				
()	解析結果の精度が高くなれば利用すると思う				
()	一つの個人情報に対して多くの視点から解析が行われるようになれば利用すると思う				
()	より解析結果が早く表示されるようになれば利用すると思う				
()	自分をもっと SNS を使うようになったら利用すると思う				
()	その他				

対象評価項目	有効性												
質問 12 privasee.me を利用していただき感じた、プライバシーに関する意識について 質問 12-1 privasee.me を利用することで、「SNS を利用した個人情報の漏えい」について関心を持ちましたか？ () 関心が低くなった () 関心は変わらない () 関心が高まった 質問 12-2 privasee.me を利用することで、今後の SNS の使い方や設定の見直し、個人情報の漏えいに気をつけるきっかけになりましたか？ 質問 12-3 privasee.me を利用することで、今後の SNS の使い方を検討したり、設定を見なおそうと思いますか？													
	<table border="1"> <thead> <tr> <th></th><th>1</th><th>2</th><th>3</th><th>4</th><th></th></tr> </thead> <tbody> <tr> <td>使い方を検討しない・ 設定を見なおさない</td><td>()</td><td>()</td><td>()</td><td>()</td><td>使い方を検討する・ 設定を見なおす</td></tr> </tbody> </table>		1	2	3	4		使い方を検討しない・ 設定を見なおさない	()	()	()	()	使い方を検討する・ 設定を見なおす
	1	2	3	4									
使い方を検討しない・ 設定を見なおさない	()	()	()	()	使い方を検討する・ 設定を見なおす								

対象評価項目	有効性
質問 13 privasee.me に関して、良かった点あるいは悪かった点など、気がついたことがありましたら、自由にお書きください。	

対象評価項目	有効性
質問 14 privasee.me やアンケートに対して要望・質問・意見などありましたら、自由にお書きください。	

SNS からの意図しない個人情報漏えい検出システム
アンケート結果まとめ



目次

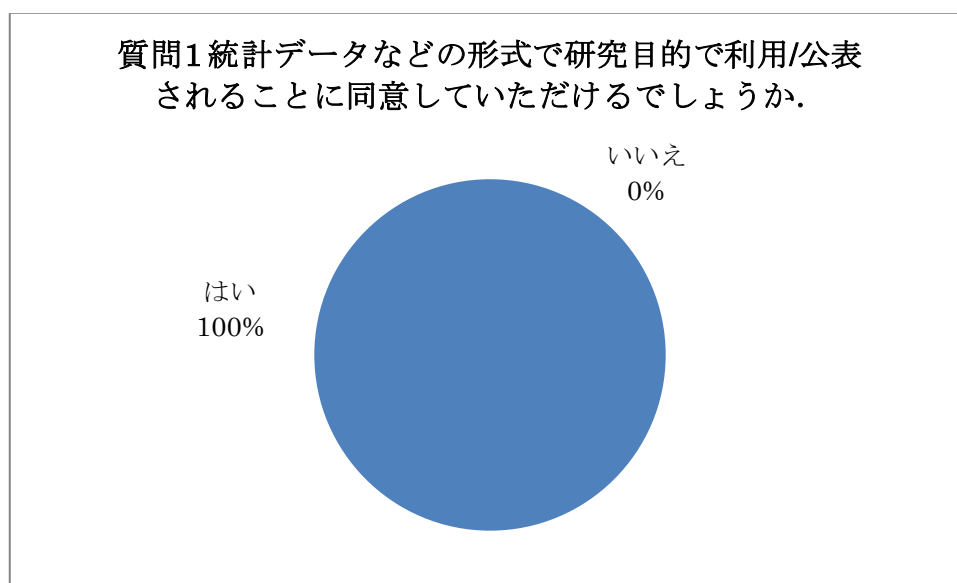
SNS からの意図しない個人情報漏えい検出システム	77
アンケート結果まとめ	77
アンケート実施概要	79
質問 1. アンケート協力依頼について	79
質問 2. 利用者の SNS 利用状況について①	80
質問 3. 利用者の SNS 利用状況について②	81
質問 4. ユーザインタフェースの分かりやすさについて①	83
質問 5. ユーザの主観的満足度について①	85
質問 6. ユーザインタフェースについて②	86
質問 7. 解析の正確性と効果について①	89
質問 8. 解析の成果性と効果について②	90
質問 9. 解析の正確性と効果について③	92
質問 10. 解析の正確性と効果について④	92
質問 11. ユーザの主観的満足度について②	96
質問 12. 有効性について①	98
質問 13. 有効性について②	99
質問 14. その他	102

アンケート実施概要

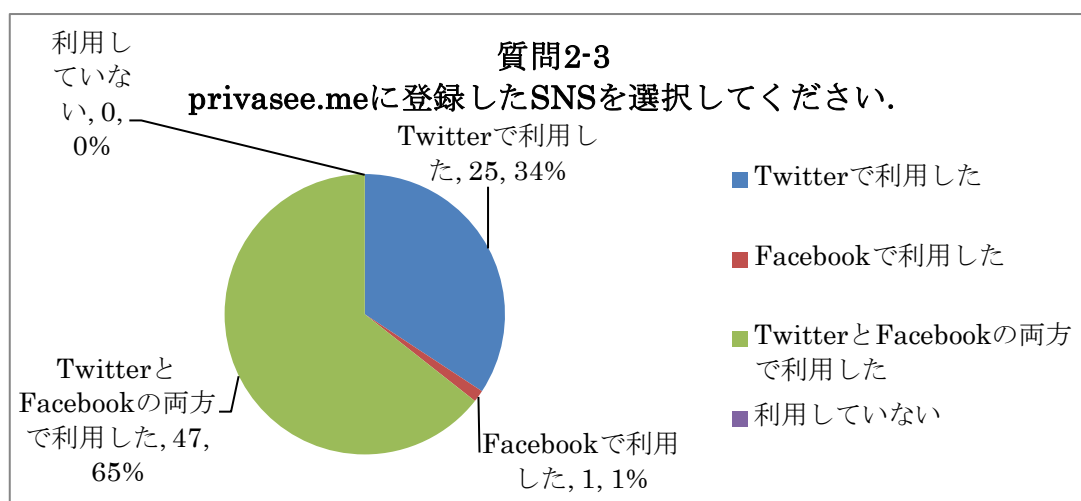
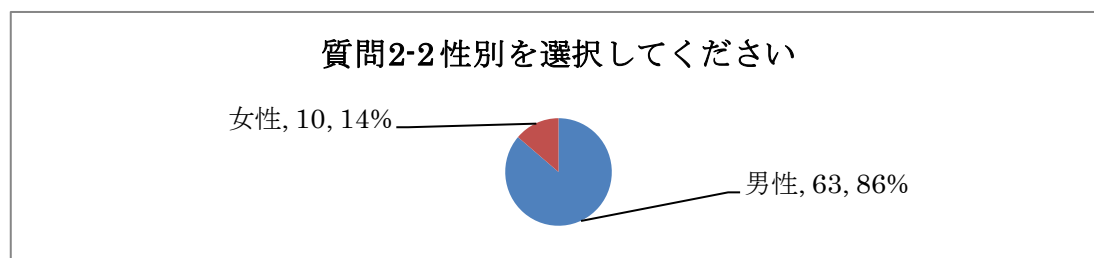
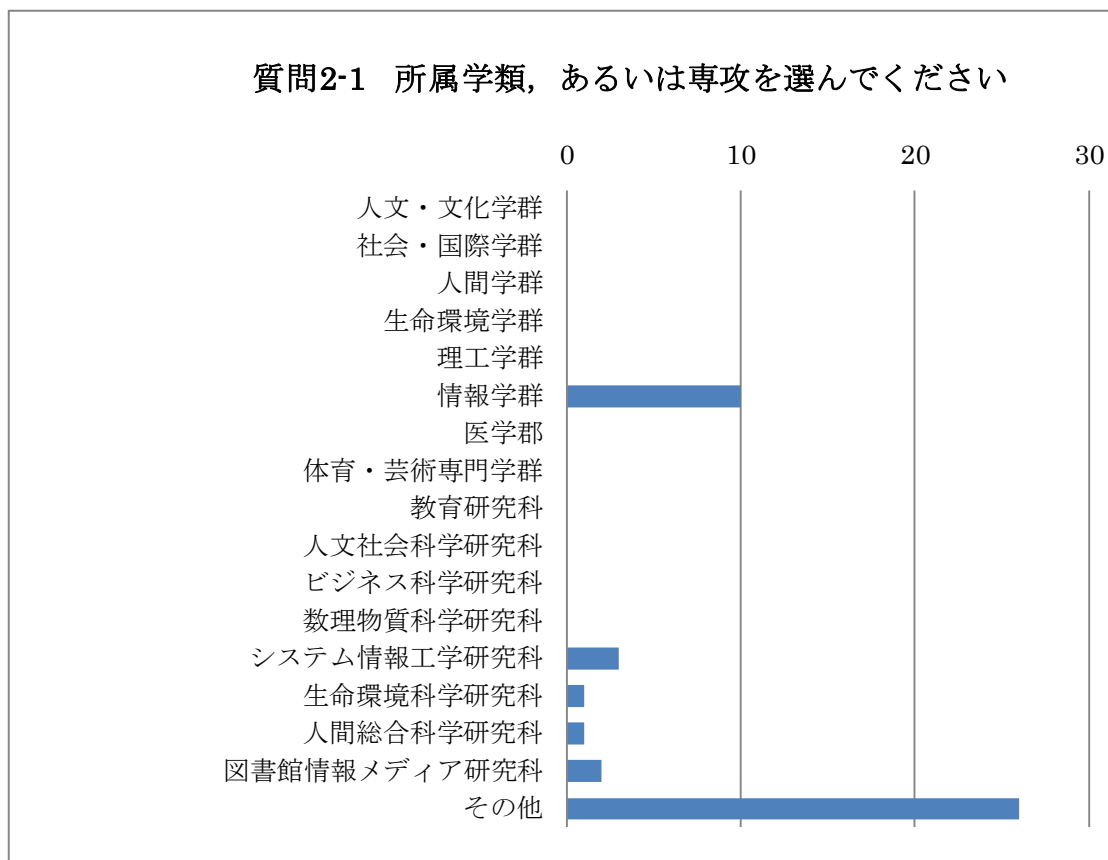
アンケート実施概要は以下の通りである.

・有効回答者数：73 名

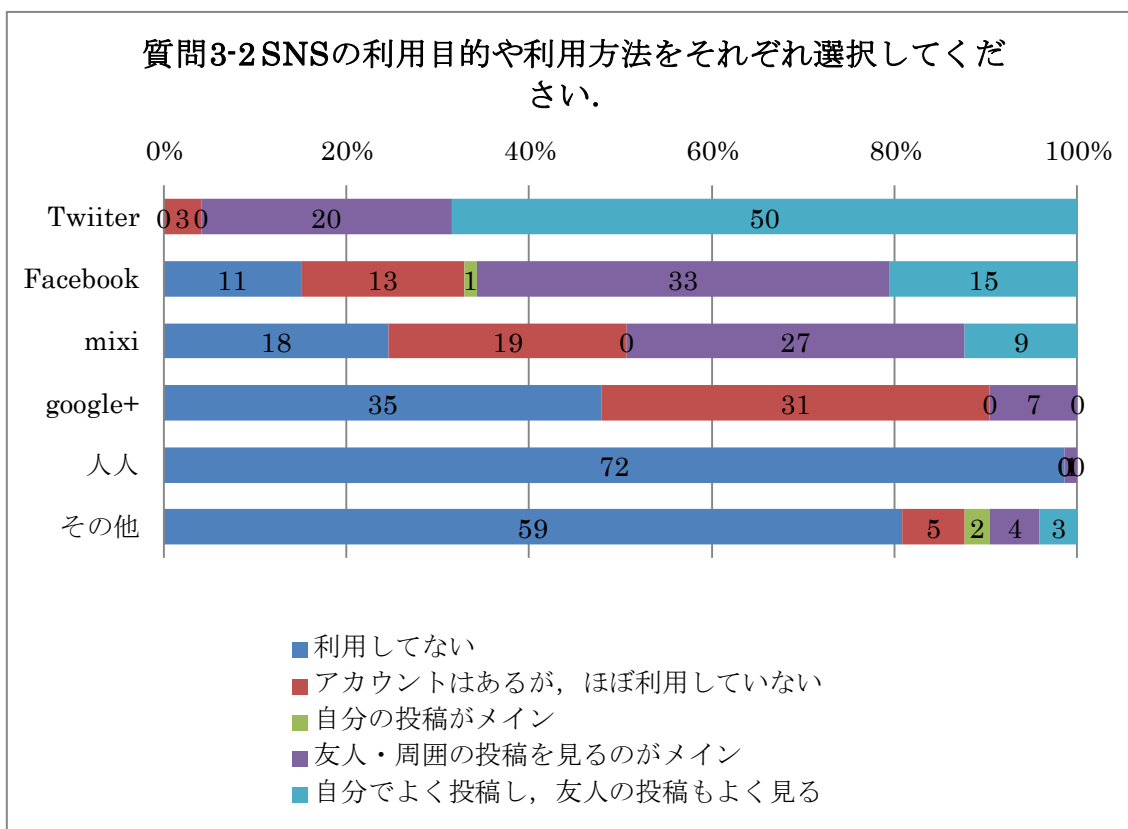
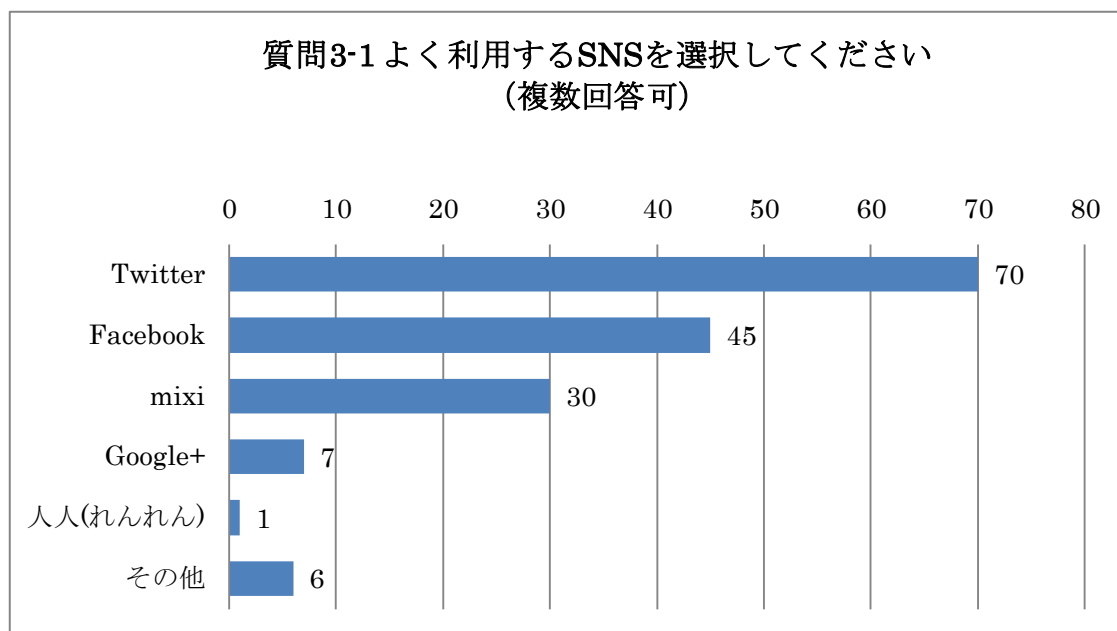
質問1. アンケート協力依頼について



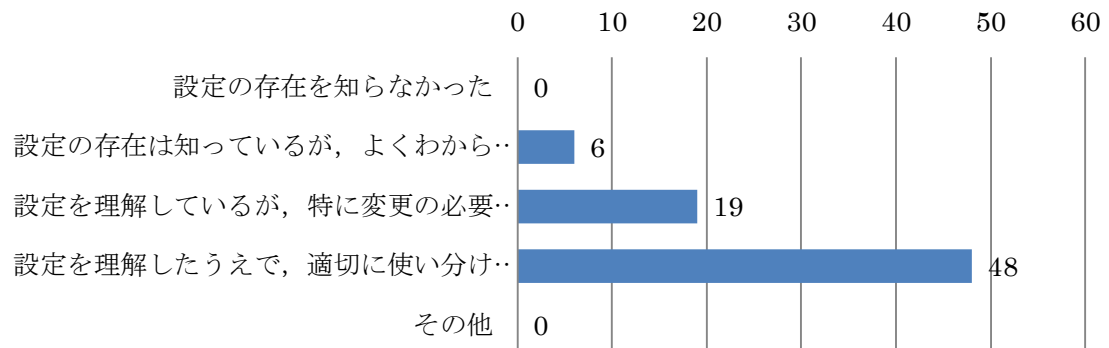
質問2. 利用者の SNS 利用状況について①



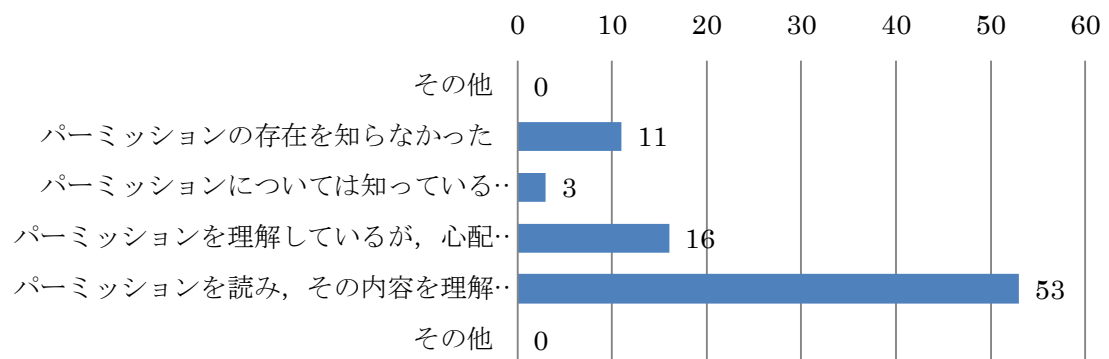
質問3. 利用者の SNS 利用状況について②



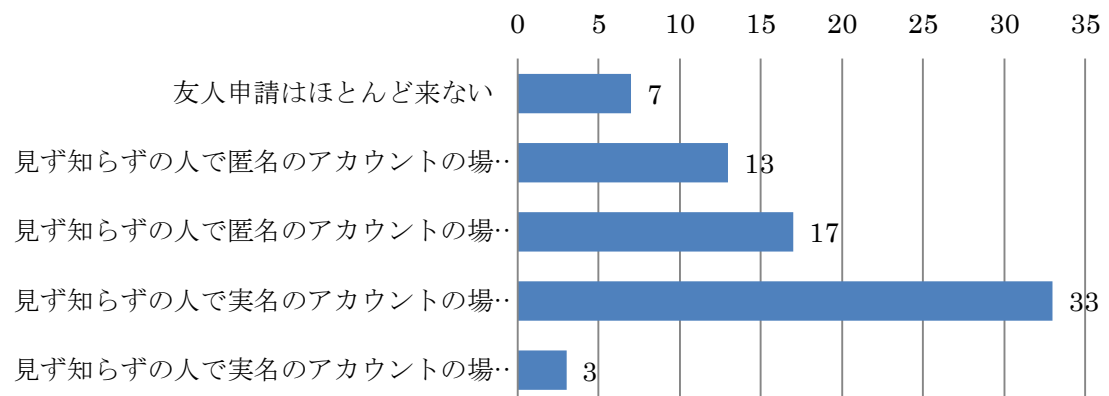
質問3-3 あなたは設定を理解し、適切に公開範囲を定めていますか？



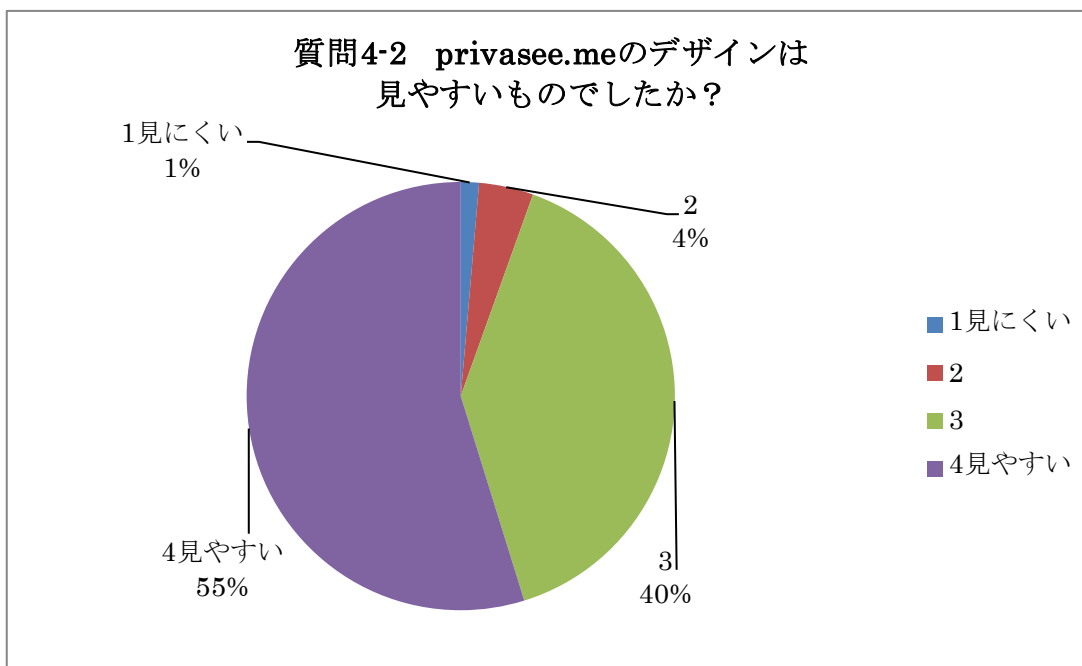
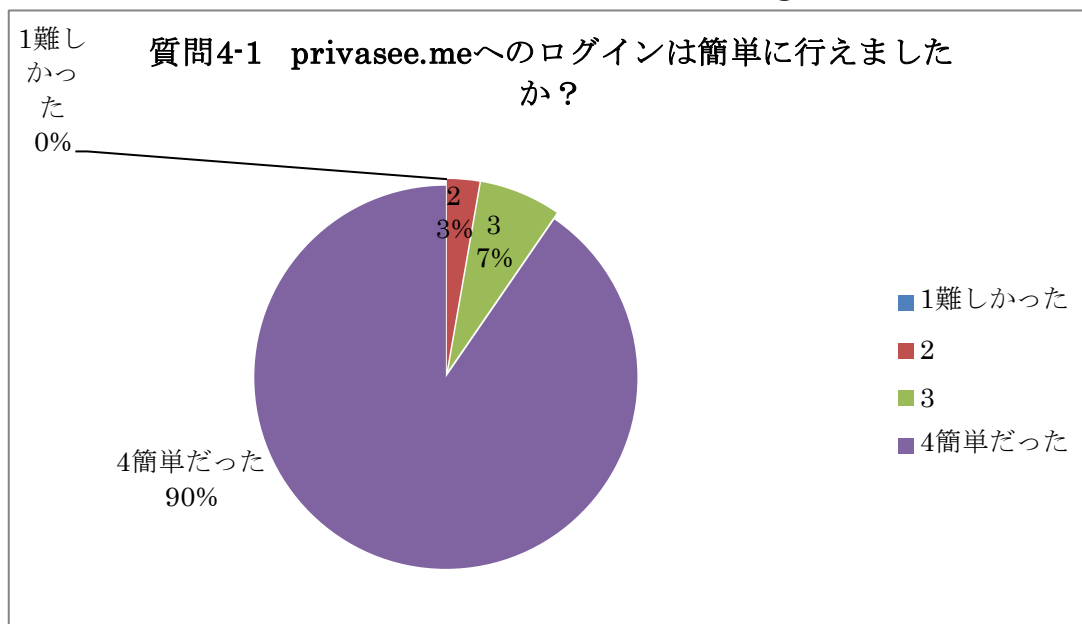
質問3-4 あなたはパーミッションを確認して、アプリケーション連携の認証を行っていますか？(複数回答可)

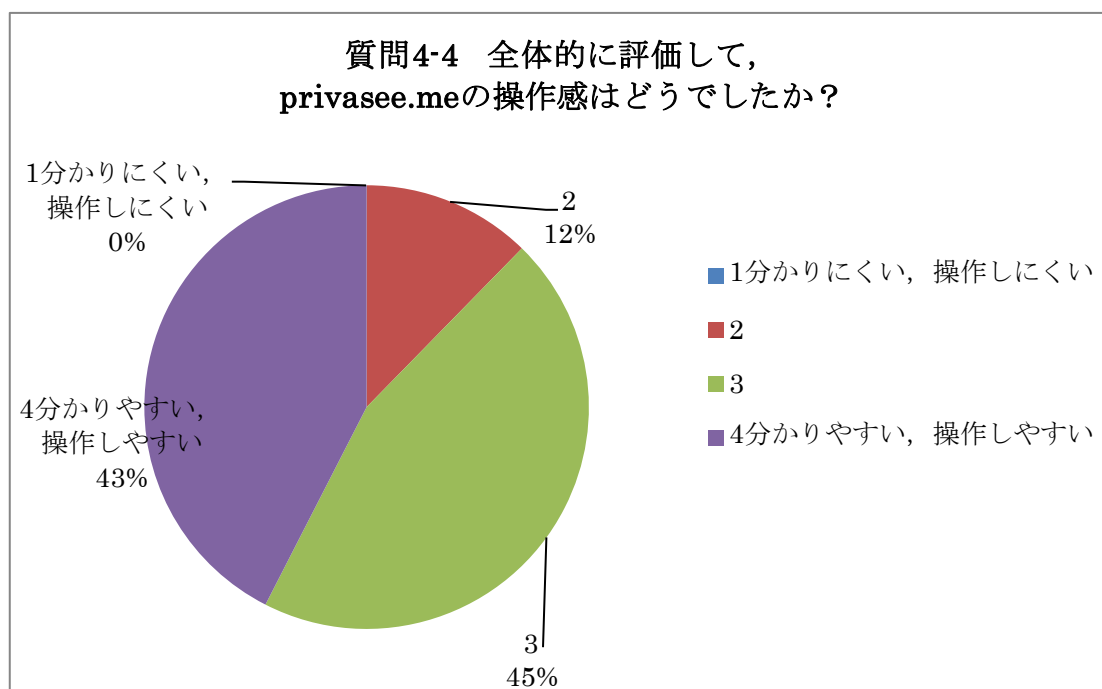
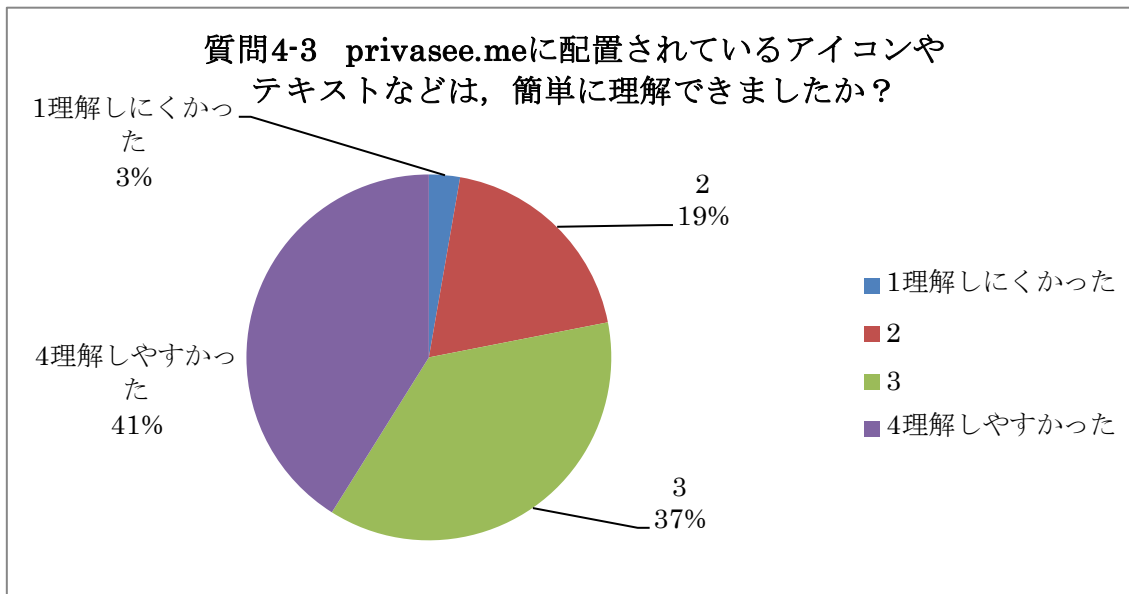


質問3-5 知らない人からの友人申請が来た場合について、あてはまるものを選択してください。(複数回答可)

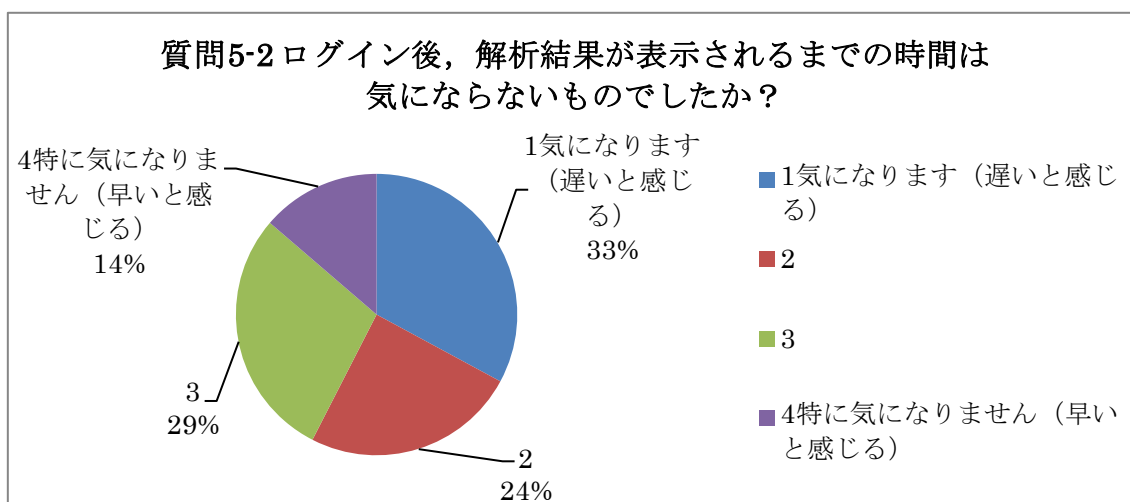
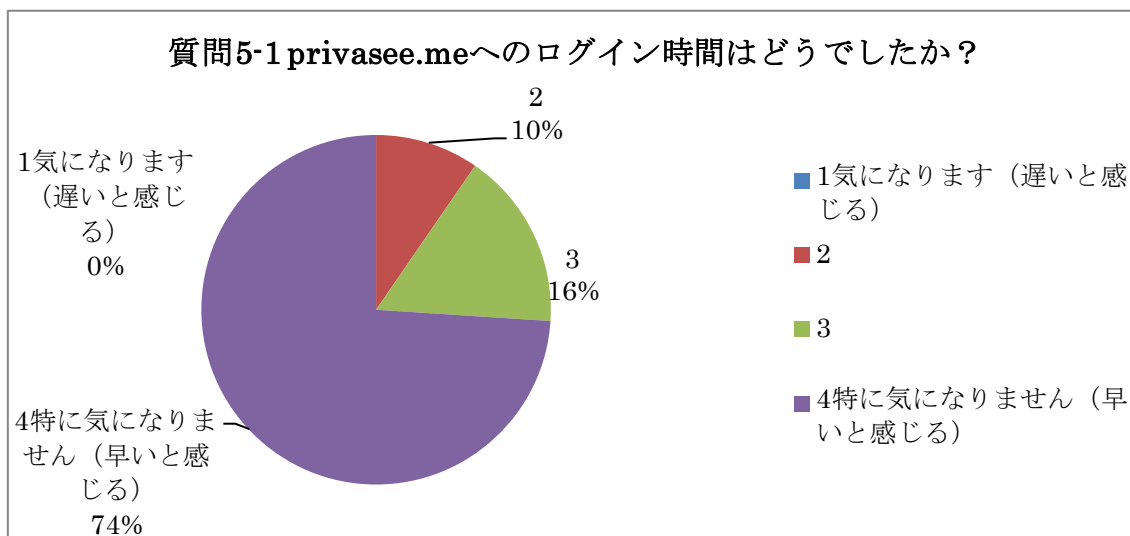


質問4. ユーザインタフェースの分かりやすさについて①





質問5. ユーザの主観的満足度について①



質問6. ユーザインタフェースについて②

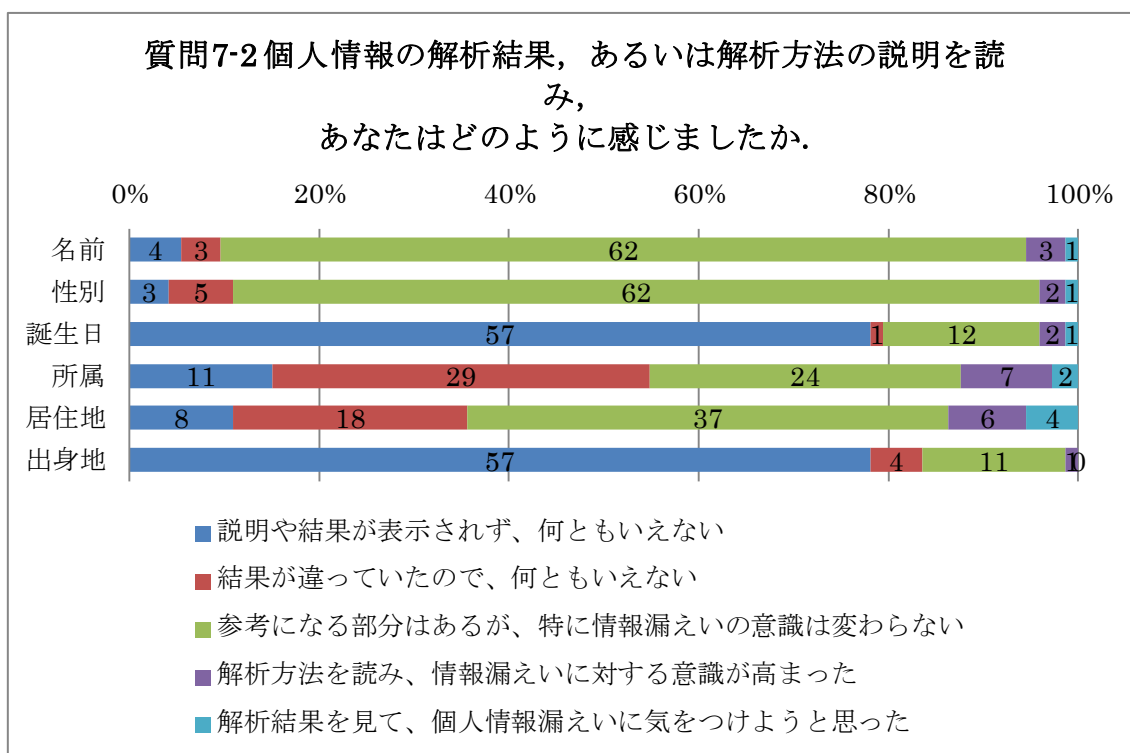
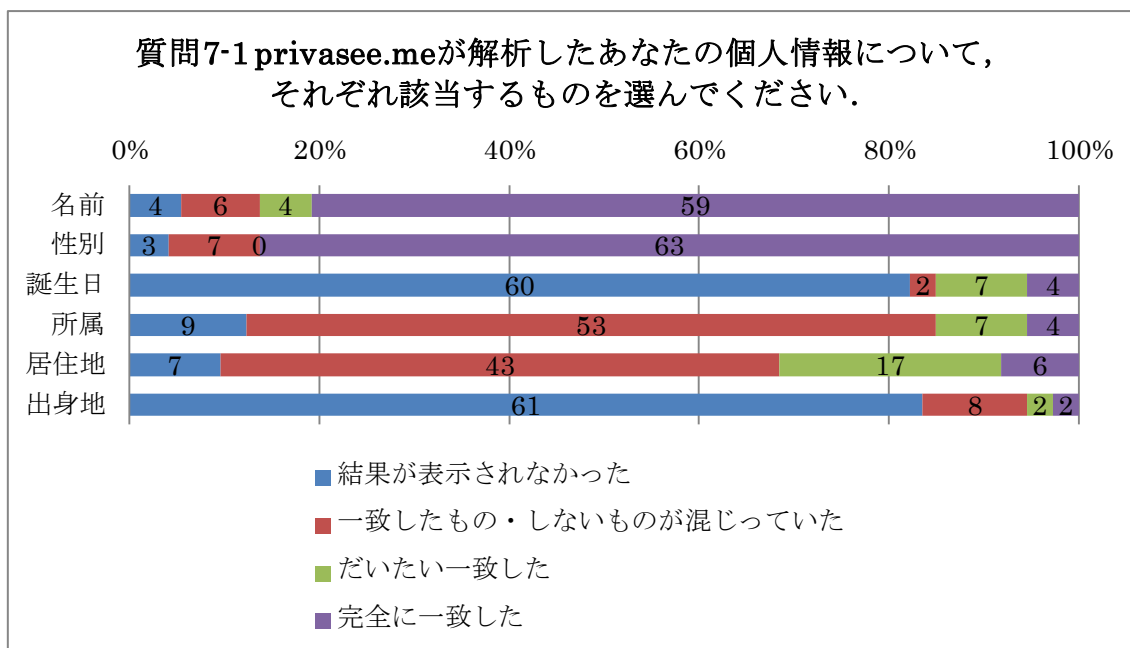
- 調査結果の表示の際に画面遷移があると操作感が落ちると思う
- 調査結果の表示のデザインと、トップページ?のデザインに統一感があると良かった"
- トップ画面右上のアイコンでログインに利用した SNS がわかるようになっているようですが、わかりにくいです (モノクロとカラーだと判別が付きにくい)。解析結果が表示されるまでどのくらい時間がかかるかわからないので、正しくログインできているのか不安になります。自分で画面を更新しないと解析結果が表示されないようですが、その点に関して説明するテキストがなかったと思います (メールには記載されていましたが)。
- シンプルだがサービスを利用するための要素が適切に配置されていて良いと思います。解析結果を閲覧するまで特に迷うことがない作りになっていて、IT サービスの知識や経験が少ないユーザーでも簡単に利用できるように配慮されているように感じた。ベストフレンド解析などで、結果が 0 の場合は何かメッセージを表示したほうが良いと思いました。 「<table border="1" cellspacing="0"></table>」 だけが表示されていました。
- 所属についての解析結果がよくわからなかった (謎の文字列が並んでいた) 解析が終了したことがわからない。表示される項目がどれかをわかるようにして欲しい。分析結果をみんなで教えあうのが面白かった。結果を twitter でつぶやけるようにすればよいと思う。
- 2点不満に思う点があった。
 - 1.解析結果の読み方に関して
頻出表現解析に関して、提示された情報から何をどう受け取れば良いのか悩んでしまった。
頻出表現解析の結果提示ページでは以下のような文章が提示されるが、提示されたの語に続く ""(4)"" など数値の意味をどう汲み取るべきか把握する事が出来なかった。
>ORGANIZATION:筑波大(4)
>PERSON: マジ (7)
解析を行わせる前に、一度利用の流れをユーザに提示するべきであると考えている。現在の結果提示方法に関して、非常に強い不満を覚えている。
 - 2.privasee.me のベストフレンド解析に関して
画像を投稿していないユーザに対する解析結果の提示に不満を抱いた。
画像が無い場合は解析が出来なかった旨をユーザに提示するべきではないか。
>Facebook の写真をみると、あなたが誰と仲が良いかが他の人にバレるかもしれません
結果提示のページに書かれたこの文言だけでは、解析が現在も行われているのではないかと感じてしまう。
- 解析結果に明らかにおかしい単語が含まれている。ツイッターのみでの登録だが、解析結果が表示されない誕生日や出身地の項目は、トルツメにするか結果が存在しない旨を記述すべきだと思った。最下部の 4 つのリンク先がページによってバラバラでおかしい。単純なバグだと思いますが... 下のコピーライトとリンクはグレイアウトされてると選択できないのかと思うので、白いページ内に入れるべきだと思う。
- なぜその結果になったのかの理由の提示が欲しかった。所属という表記が良くわからなかった
- 結果を自分でリロードしなければならないのがちょっと面倒。それ以外は特に問題ないと思います。 "
- 錠前 (?) のアイコンがサービスの性質をよく表していてよかった。
- 質問の意図からはずれるかもしれないが、性別判定の理由が知りたかった。

- 右上のボタンがトップページだとログアウトなのに、他のページに行くとトップに戻るになっているのはわかりづらいかなと思う。戻るボタンを探してしまった。
- 1. 特定の政党に所属していないのに民主党所属になっていたのが気になった。2. 居住地に「朝」が入っていたのが気になった"
- "良かった点：
- 画面レイアウトがとてもシンプルで見やすい点
- 悪かった点：
- 解析がいつ終わったかが分からない点。
- (どのタイミングで画面を更新させるべきが分からない)
- どのタイミングで、更新ボタンを押せばいいのかよくわからなかった。
- 結果はおもしろかったですが、既存の項目では役立て方が思いつきませんでした。居住地については、いったことのないシドニーがでてきたのが不思議でした。
- 全体的にデザイン（この場合「装飾」の意味ですが）レベルが高いのに驚きました。
- 解析器の名称が、～解析だったり、～解析器だったりしていて、統一されていない印象を受けました。
- twitter, facebook のアイコンで情報の出元を示す UI はシンプルで見やすく、意味もわかりやすいと思います。
- 頻出表現解析器の結果表示が何を意図しているのかわかりにくいと感じました。表形式は不適切かもしれません。
- それに対し、性別判定の結果表示は、「あなた男性と推測されました」と明確でいいと思います。（男性 or 女性だから明確にできるのであって、他の結果表示では難しいのかも知れませんが）
- 誕生日ツイートで誕生日がいつであるかが漏洩してしまうということは、意識していなかった。しかし、これを防ぐのは難しそうに思う。防ぐ術を参考までに提示してくれば良いのと思った。（他の情報についても、恐怖を煽るだけではなく、具体的な対策にまで踏み込んで解説があると良いのでは。）
- 名前はシンプルでいて、サービスをよく表現できていていいと思う。操作についても特に難しいと感じることはなかった。デザインについても、危険性を想起させるような配色と感ずるので、いいと思う。ただ、私の知識が足りないのかもしれないが、解析結果画面の人形の画像？を使う意図がわからなかった。"
- トップの人形の画像の意味がよくわかりませんでした
- いい idea を組んだアプリだと思います。
- 左上のアイコンクリックでホームに戻れた方が便利。結果画面のトップに戻るとログアウトが同じ場所にあるのが操作ミスにつながりそう。頻出表現解析器で 1-5 位と上に表示した方がわかりやすくいいと思う。
- 解析中の状態はアニメーションなどにしたほうが分かりやすかった。文字だともう結果出たのかと思ってしまった。解析結果はすべて 1 ページでもいいんじゃないかと思った。
- やってみて面白かったです。ユーザの SNS 利用スタイルを入力させ、分析結果から適切なアドバイスをいただけるようなシステムがあると面白いと思いました。
- 頑張ってください。
- 非常に見やすく、使いやすいツールでした。また、デザインに関しても、随所にセンスが光るもので、とても感心しました。（アイコンもいい感じです！）
- 同じ学生が作ったとは思えないクオリティで驚きました。"
- ベストフレンド解析など、よくわからない数値がそのまま出す必要はないのではないかと思った。フォントのサイズを変更するなど工夫ができたと思う。facebook には性別

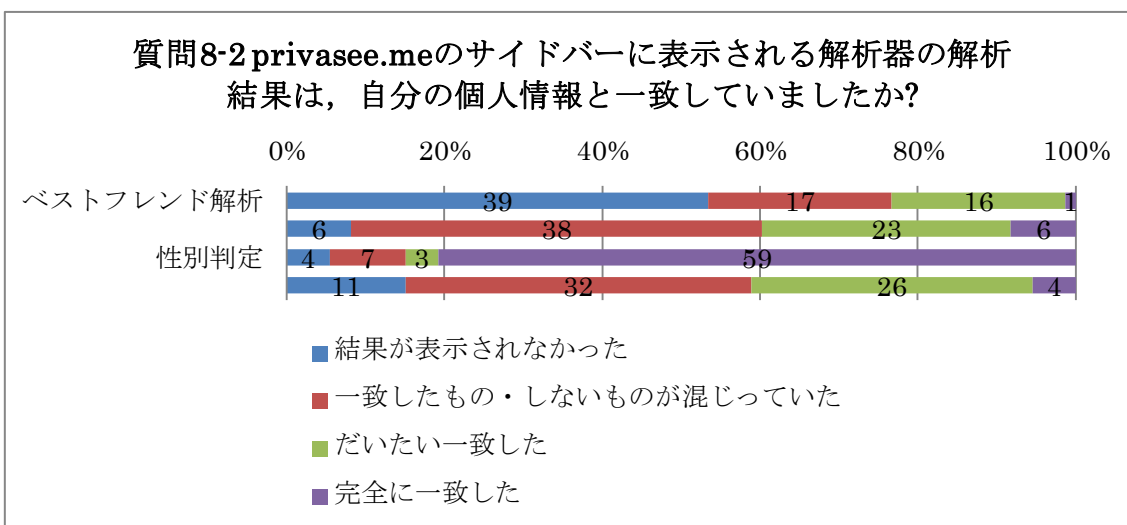
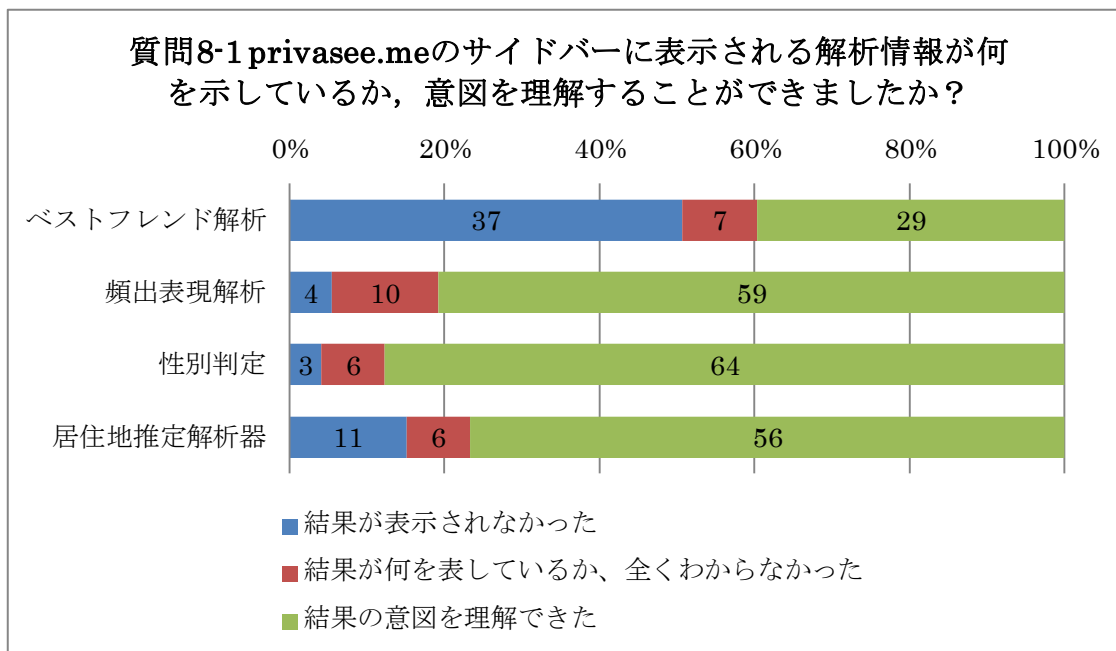
も表示しているが、性別を識別できなかった(なぜ?). また facebook の居住地をみれば場所などわかるのに、オーストラリアのシドニーを指している(twitter でもオーストラリアだった), よくわからない. ベストフレンドに限ってはまあまああたっているかも解析結果がすべて一緒に表示されるのは少し見づらかったように思う。

- ログアウトのリンクがあるのにログアウトできないのはどうかと思う。Sleipnir3(IE9)で地図の表示がされなかった。"トップページのロゴの下にある画像が何を意味しているのか気になった。頻出表現解析器に関しては、「関係組織・人物相関」のようなタイトルにした方が自然なのではないかと感じた。頻出表現から推測していることは説明文に入れるなどの方法もあると思う。何か意図があったとしたらごめんなさい。
- 普段使っている SNS の情報から”情報漏えいの可能性”を可視化するというのはとても興味深いコンセプトでした。このような実験に参加させていただきありがとうございます。
- デザイン。工学の人が見れば十分だと思うが、一般の人が使うには少し固いデザインだと思う。
- 解析に時間がかかるのであれば、どの程度かかるのか表示して欲しい。
- 解析中(now crawling)から表示が切り替わるタイミングが分からず、どうしてよいかわからなかった。(リロードすべき, ひたすら待つべき, スクリプトが止まっている etc...) ちなみに、利用環境は Firefox 3.6 でした
- 解析結果を、専用のページに遷移して(性別の結果や、所属の結果など)観ることができたが、大きな画面に少しの情報しかないため、少しさびしい感じがしてしまいました。▽を押下すると結果のページが展開される形式(うまく説明できませんが)でもいいのかと思いました。
- トップページは情報がまとまっていてわかりやすいと思いました。ただ、表示されている情報がなんなのか説明されていないのが気になりました。空気を読めば解析結果から予想するプロフィールが表示されていることがわかるのですが・・・
- また、ベストフレンド解析などの詳細な解析結果の数値等の意味が書かれていないのでよくわからなかったです
- 解析をスタートしてから解析に時間がかかることがわかってテンションが下がったので、最初に説明があるとユーザーにやさしくなると思います。
- 全体のデザインは、今風でとても良いと思います。
- 更新ボタン、解析終了の表示などがあるといいと思う
- 地図がでてくるものが非常に面白かった。可視化されていて面白いと思う。
- 更新を押さなければ結果が表示されなかった点が少し気になりました。
- GoogleMap にプロットされてるのはわかりやすいと思いました。解析器の詳細なデータが出るのも有り難いのですが、多分自身が発言した単語を拾ってきているのだと思いますが、どの投稿から拾ってきた単語なのかが分かると、もっと良いかと思います。既に実装されていたら申し訳ないです。

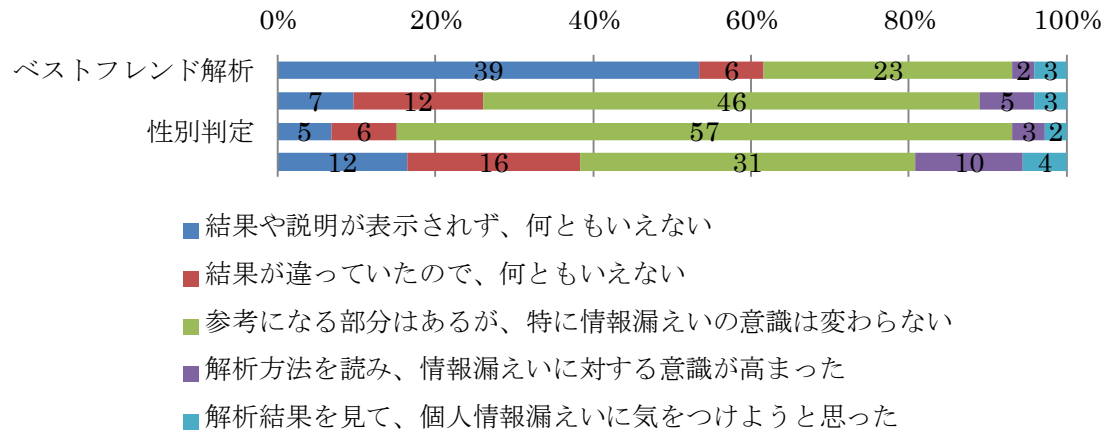
質問7. 解析の正確性と効果について①



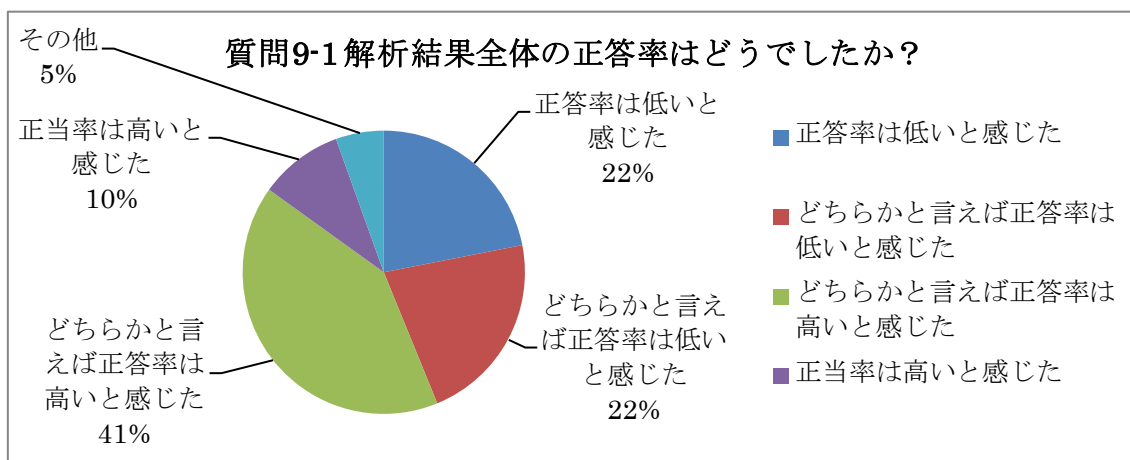
質問8. 解析の成果性と効果について②



質問8-3 privasee.meのサイドバーに表示される解析器の説明
文や解析結果を、どのように感じましたか。



質問9. 解析の正確性と効果について③



質問10. 解析の正確性と効果について④

- Facebook の「いいね！」の数や、Twitter の@の数の統計なんかからデータを出してくれと、身近に感じるかもしれない
- 解析結果がどうなったときに、どのようなリスクがあるのか分からないため、対策を取る必要があるのか判断できない。また、解析結果にリスクを感じたとしても、どのような対策を取るべきかわからない。
- 「言葉遣いから性別が推測される可能性があります」とありますが、原理がわからないので対策のしようがないです。私の場合は居住地に「ダイソー」と出ましたが、友人の場合は所属に「ダイソー」と出ていました。居住地推定解析器のマップに関してです。私の場合、通学先が一番赤く表示されました。また、居住地には何もマークが付きませんでした。各マークの平均位置を取ると（明らかに外れているマークは除く）およそ居住地を示すのではないかと感じました。
- 「他人から漏れた情報」が切り分けられていないように感じたので、他人から漏れた情報はどれくらいか示してくれるとわかりやすかったと思います（Twitter の@のみで解析した結果とか？）。
- 居住地は何番地とか何号室というところまで特定されないと脅威と感ない。「居住地推定」ではなく「活動範囲推定」などに言い換えたら良いと思う。
- 居住地に居住地が出ず、出身地が出てきたので、居住地と出身地がどのように分けてあるかが分からなかったです。結果が出なかった項目は、出ませんでしたと書かれてたほうがよいと思います。
- 私は普段自分の場所については話さず、旅行先ではツイートするので、頻出単語はほとんど旅行先になったように思います。
- Facebook のベストフレンド解析は、基準がわかりませんでした。Facebook のほうが個人情報が載っているのに、項目が twitter に比べて少ないと思います。
- 私は Facebook では自分の所属などを公開しているので、公開されているプロフィールに関してはこれに表示してもいいかと思います。
- 学生 or 社会人を判定できると面白いと思う。
- 解析結果に"マジ"(形容動詞)、“アレ”(指示語)など予め解析対象から取り除くべき語が混じっており、解析の手順に問題があるのではないかと感じた。
- 正答率が低い。
- もともと特定されるような発言をあまりしないので仕方ないのかもしれない。

- 試みは興味深いとは思ったが、精度や理由が不十分であった。なぜその解析結果になったのかが表示されたら良いと思った。
- 解析失敗例：男性なのに女性表示、頻出表現が「ー」（長音）、所属が「ソフトバンク」
- 大まかな居住地などは公開しても良いと思って公開しているので、問題には感じなかったが、意識していない人が見ることを考えると、有用だと感じた。
- 自分も、「公開したくない」範囲の、詳細な情報が解析されていたら意識を変えさせられると思う。頻出表現だけでなく、そこから類推した情報が提示されると、効果が上がると思う。また、ORGANIZATION の精度の低さは問題だと感じた。（4/5 間違っていた）
- 良かった点
 - ある程度あたっている
- 悪かった点
 - 性別判定の理由がわからないので、仮に情報漏洩に気を付けようと思っても、どうしたらいいかわからない
 - データが少ないと解析結果がいまいち。自分の場合は Facebook であまり写真に写ってないのでベストフレンド解析の結果が 7 人出てみんな 0.25 だった。どのぐらいの値があれば親密なのかという基準が表示されていなかったのも、ピンとこなかった
 - 自然言語処理をやると仕方ないと思うが、地名ではないものが地名になっていたり、人名ではないものが人名として処理されていた。
- 文章解析で、reply 相手のアカウント名が入っていた。これを利用してベストフレンドも出来たのではないかなと思うのですがどうなのでしょう？ベストフレンドが facebook しか利用されなかったみたいなので、書いてみました。
- 頻出表現解析器の中で PERSON の項目の単語が何に使われるのか分からなかった
- 自分の発言や情報開示はとても少ないです。他人の情報を見るだけです。
- どのような解析をしているかは分かりませんが、私は良いサンプルではないと思います。またアカウント作って誕生日を迎えてないからなのかもしれませんが、誕生日が表示されませんでした。
- 所属に関して、わざと文脈的に言葉をカタカナにしたときのワードがでてきました。居住地に出身地がでてきました。それと、OO の食べ物がおいしかった。といったときの旅先のワードもいくつか。
- 出身地推定は、twitter だけなら日本の九州だと推定されましたが、海外のユーザとの連携の多い facebook を追加すると、出身地がオーストラリアになりました。自分自身、google の検索等にひっかからないようにしたりとか、出しても実名原則の facebook での実名程度なので、あまり出てこなかったのかもしれないです。
- ツイッターを英語で使ってるせいなのか、誕生日は 1 日早い日が表示されました。大学名を取得できてるのに、なぜかその大学から数百キロ離れたところを居住地と推定されたので、大学のロケーションを使ってみてもいいと思いました。誕生日以外は総じてすぐわかるような情報だったので、ドキッとするようなものではなかった。質問 8 の「privasee.me のサイドバーに表示される解析器の解析結果は、自分の個人情報と一致していましたか？」における頻出表現解析器については、普段発信している情報の傾向を自分で把握していないので、結果を見ても自分の個人情報として一致しているかどうかはなんとも言えず、「ああ、こんなこと発信してるのか」と思うしかなかった。
- Twitter 上の頻出表現の解析はいろんな研究で散々行われていますが、(勉強会などで何回も見ました、自分もやったことがあります) 今回の privasee.me で用いている頻度解析(および頻度解析の分類)は、ややピントはずれな気がしました。
- 地名の細かさがバラバラですし、所属も PERSON (関連深い人を表す欄でしょうか?)

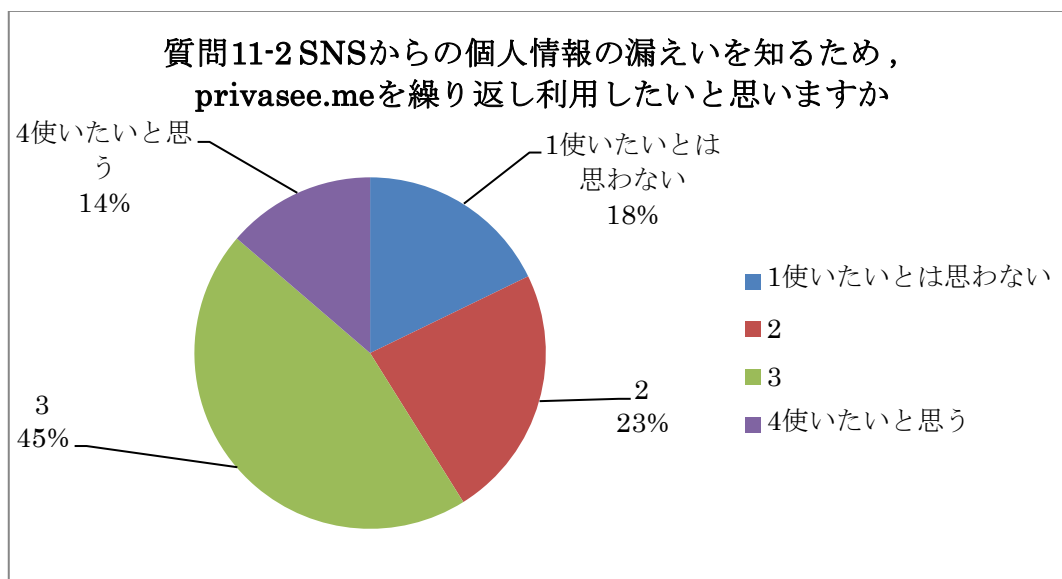
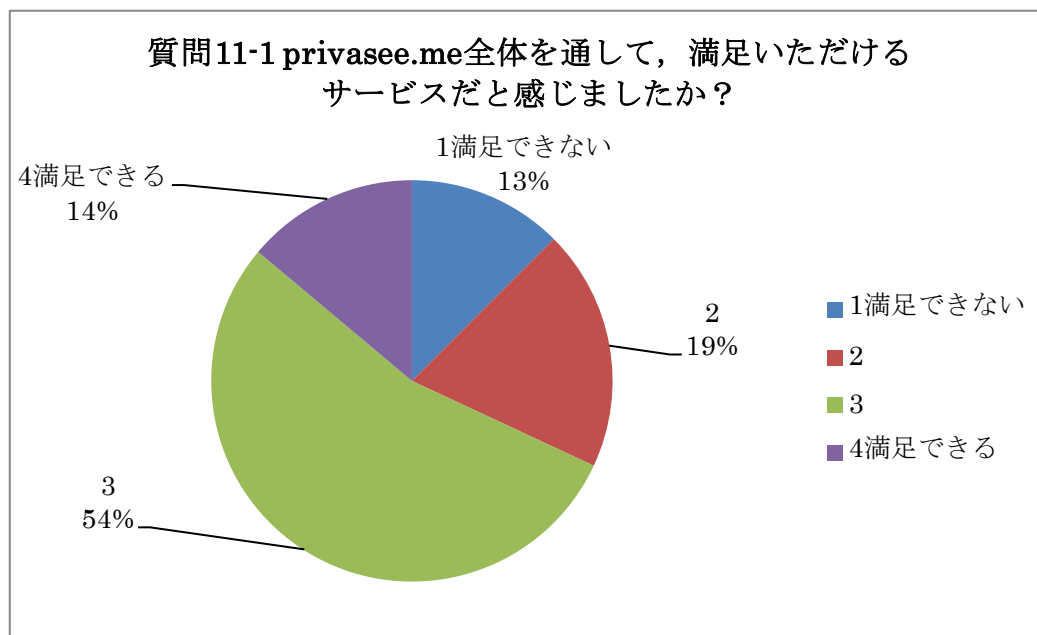
も、出したようなデータはわかったのですが、出したようなデータが出ているとは思えませんでした。元のデータ（Twitter の書き込み）がそうとう汚いもの（のはず）なので、解析器がクレンジングを注意深く行う、あるいはクレンジングをうまく自動化（ノイズ同士が打ち消し合うような仕組みを作る等）すると、もっとよい結果になるのではと思いました。

- 解析結果は、もっと噛み砕いた表現で示していいと思います。例えば、頻出表現解析器では、「○○という LOCATION に関する表現が○回現れている」という情報を表示するよりも、「あなたの居住地は、○○か××か△△であると推測されました」と表現した方が、解析の意義を感じやすいと思います。
- 私は facebook で、性別や誕生日、居住地等、基本的にすべての情報を公開しています。つまり、完全に漏洩しています。しかし、今回の解析では、facebook からこれらの情報が漏洩している旨が表示されませんでした。
- （意図的に公開しているから、漏洩でない＝表示する必要がないという判断なのかも知れませんが）他社に漏洩しうる個人情報を出すのがサービスの目的ならば、これらの情報也表示すべきだと思います。
- 居住地の結果が適切に地図のほうに反映されていなかったように感じた。
- 自然言語解析でこのようなことができるのは、すごいと思いました。着眼点が面白いシステムであると感じました。プライバシーがなにかと騒がれ、Twitter での不用意な発言による炎上が頻繁に発生している現状において、需要があるシステムだと思います
- 開発によく頑張ったことが見られると思います。
- いいアプリのではないのでしょうか。"
- 居住地推定解析器で関東とか九州とかで中国(国)の場所が表示されたりしたので、地点だけで無く、地域についても表示を考慮出来たらいいと思う
- おもしろかったです。
- iphone のアプリから投稿されたツイートの単語が頻出表現にひっかかっていました。
- 先に Twitter を認証したことに所以するのか分かりませんが、facebook からの解析がうまくいってないと感じました。（個人ページの各欄に表示されるアイコンが殆ど Twitter のみなのですが、これは仕様でしょうか？今回は表示されなかった誕生日も、本来ならば facebook から取得する予定だったのではないかと思います。）その他の解析については、ベストフレンズの解析結果に全く知らない人も多々登場しておりました。上記はさておき、非常によくできたシステムでした！"よく趣旨が分からなかった
- 居住地推定解析機がエラーをはいてしまい見られなかったのが残念でした。
- 性別は、自分の性と違うものが判定されたのですが、これはきっと Twitter と Facebook では正確性がちがうだろうなあと感じました。とくに日本の場合。あと、名前はそもそも Twitter では本名を名乗っていないのですが
- 解析結果は Twtter で登録している仮名がまるまる出て、これは個人情報と一致したと言っていいのかよくわかりませんね。一応 Twitter で名乗った名前が完全に出ているので、完全に一致したにチェックを入れてはおきましたが・・・。
- 性別の判定が間違っていた。結果がほとんど表示されなかったのが、残念だった
- 結果や説明が表示されず、何ともいえない
- 結果が違っていたので、何ともいえない
- 参考になる部分はあるが、特に情報漏えいの意識は変わらない
- 解析方法を読み、情報漏えいに対する意識が高まった
- 解析結果を見て、個人情報漏えいに気をつけようと思ったの中で「参考になる部分はあるが、特に情報漏えいの意識は変わらない」に含まれているのかもしれませんが、

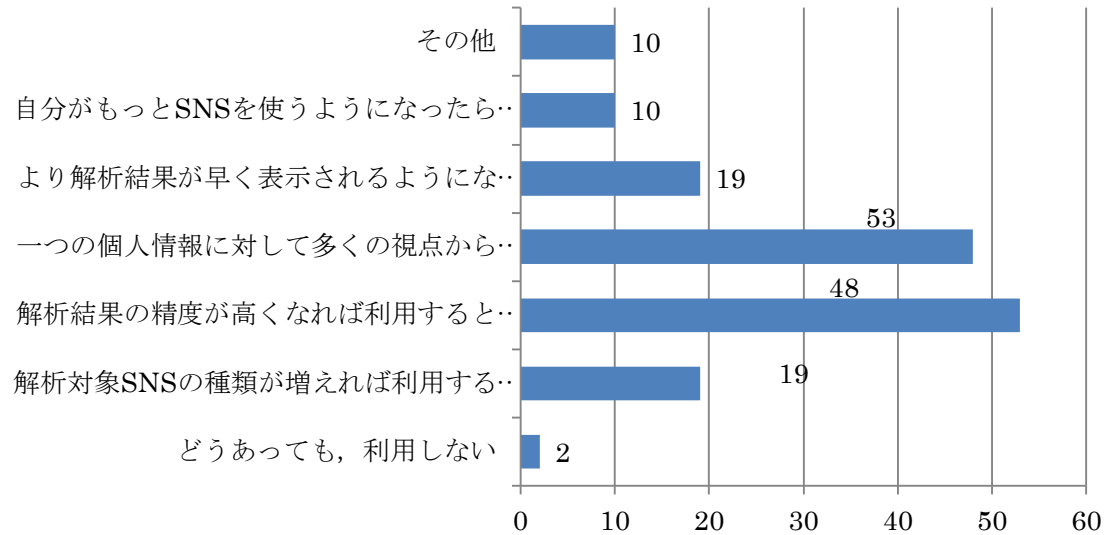
私は「参考になる部分」という意味がよく理解できず、「解析結果は合っているが、情報漏洩に対する意識はかわらなかった」と感じた。

- 私はこれら SNS に公開している情報には日頃から気を使っています。なので、回答に困りました。対象が情報漏洩に対する意識が低い人達のように思えたので「日頃から公開している情報には気をつけている」等の回答を作り、意識の低い人の割合を把握したらいいのではないかと感じた。
- アンケートの評価項目が読みづらかった。評価項目に「一致していなかった」がないのはどうかと思う。
- 性別などの推測はよくできていると感じた。(性別を明示的に公開しているつもりはなかった)居住地などは逆に、よく行く行き先が出力されたので、出現単語から推測するのは限度があるのかとも感じる。
- 全体としては非常に良い出来で、面白い取り組みだと思う。利用者視点に立った時に、どの様な利用者を想定しているのかという疑問があった。(これらのサービスを利用しようとする時点である程度個人情報に対する意識があると思うが、本当にこれらのサービスを必要としているのはそういった人たちではないのではないかと。)
- さらに、情報が特定できるならば、なぜ特定できたのかが表示されると対策の仕方が変化するのではないとも思う。(これから気をつけようだけでなく、この投稿はまずいから至急削除する、など。)
- 私は時折独特な言い回し(正しい日本語文法に当てはまらない物言い)をしているために、“所属”の解析結果がひらがな一文字の検索結果などがでており、正しくない結果になっていたのが残念でした。しかし、それ以外の解析については、ほぼ的中しておりました。
- twitter の解析に関して、1つ気になった点があります。リプライする際に、うっかり本名やあだ名を書いてしまった!・・のような場合についても考慮して解析されているのかが気になりました(@example 例題さんは~ といった状況です)。
- 解析方法をもっと詳しく説明してほしい。納得感が少ないです。例えば、過去1ヶ月間のいいねした相手を数えていますなど・・・
- 頻出表現にフレンドの名前が入っていたので、除いてくれるとより良かった
- 解析の対象とする期間によっても正答率が変わってくると思うので、最適な値を見つけることが1つの研究対象になりそうだった
- 解析結果があまりよくないことから、いまいちプライバシーの意識が高まらなかったが、データがあることから、解析精度を上げることができると思うので、今後に期待したいです。
- twitter のフォロワーからという発想は面白かった。

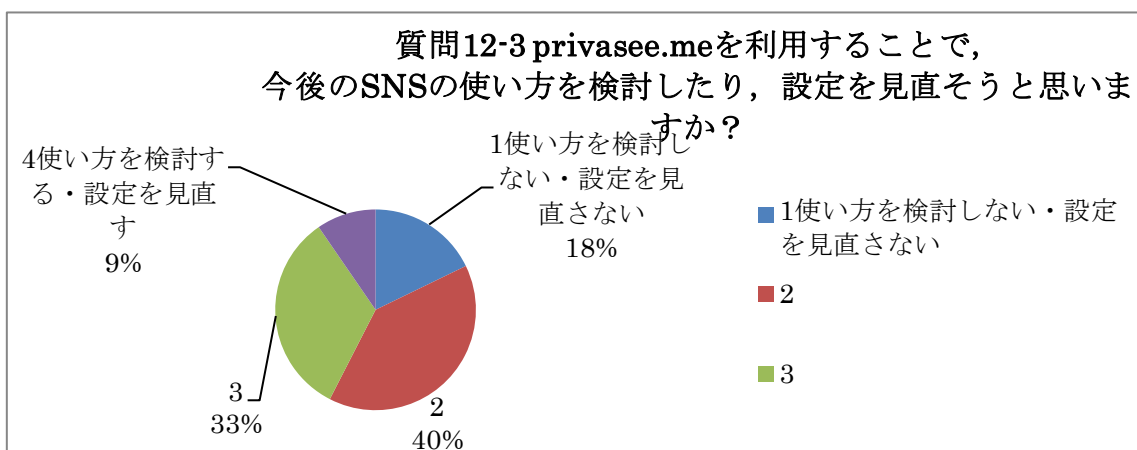
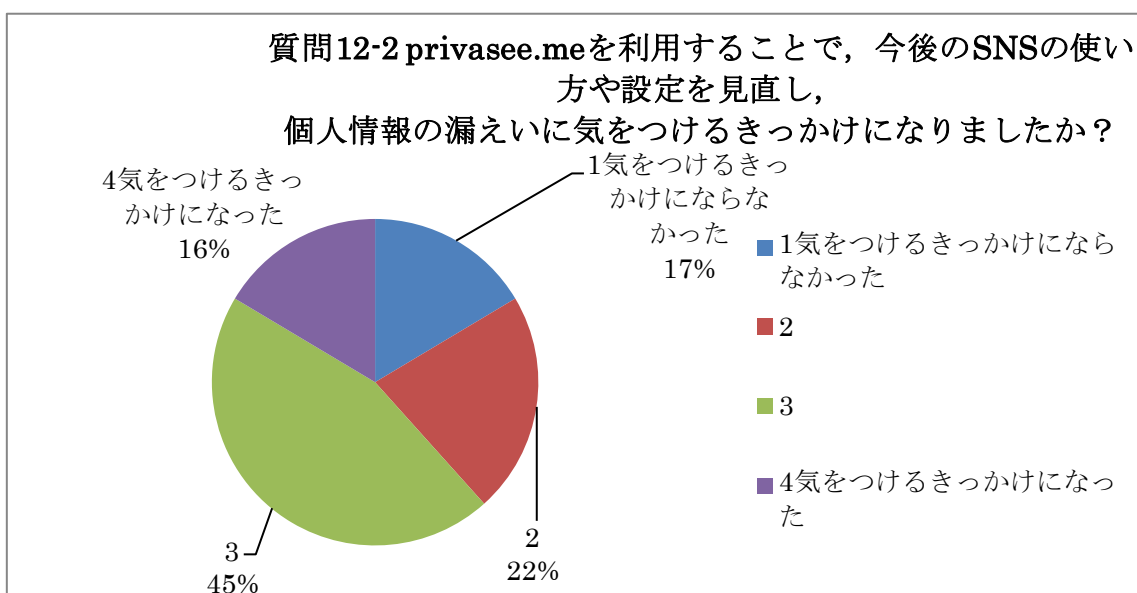
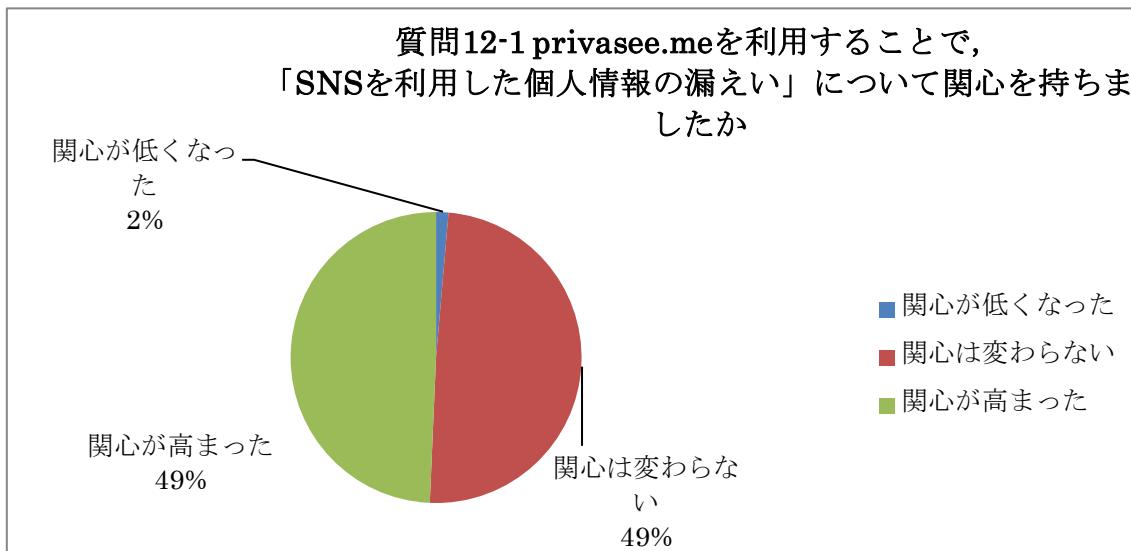
質問11. ユーザの主観的満足度について②



質問11-3 どのような改良点があると、今後も繰り返し
privasee.meを利用すると思いますか？（複数回答可）



質問12. 有効性について①



質問13. 有効性について②

- 良い点：やろうとしていることは素晴らしいと思う。需要もありそう。
- 悪い点：私の場合はすでにプライバシーには気を付けているため、この分析はあまり役に立たなかったように思う。
- 例えば、この分析ソフトの精度がかなり高いという話であれば、間違った分析結果が表示されたときに、「自分はプライバシーを守れているな」と確認でき、役に立つ。
- しかし、分析の実際にはかなり低いように感じた。
- (Twitter で誕生日や出身地を隠してはいないが、これらの結果は表示されなかった。)
- "
- 画面レイアウトがシンプルでわかりやすかったです。
- 使い方のガイドやシステムの目的がページ内に書かれていると良いと思います。
- 情報漏洩のリスクやその影響など、わかりやすくまとめたページがあれば解析結果から実際の SNS 利用方法の再検討に結び付きやすいと思う。
- 自分が大々的に公表していないはずの個人情報を privasee.me が暴くことができるようになったら、インターネットの恐ろしさを実感し、「SNS を利用した個人情報の漏えい」に気を付けたいと考えるようになると思う。今はそうではない。
- 個人情報漏洩を知るためと言うより、エンターテインメントサービスとして広めていったほうが受け入れられると思う。
- ユーザのとるべき行動の流れを提示してほしい。
- 自分の場合は、ある程度プライバシー情報を出すことには抵抗がなく、ここまでは出していいだろうというふうに普段から意識しているので、特に個人情報漏洩に気をつけようとは思わなかった。
- 上にも書いたが、よりプライベートな情報、ばれてると思ってもなかった情報が漏洩しているかがわかれば利用するとおもう。例えば、自分の顔写真、付き合っている人の個人情報、自宅の住所、電話番号など。
- また、SNS に投稿する情報は時間が経つに連れて増えていくので、より正確に多様な情報がチェックできれば、繰り返し privasee.me を使いたいと思う。
- 解析に時間がかかるのは、登録したあとメールで結果を伝えるようにすればいいと思う。登録の際、「定期的に個人情報漏えい度チェックレポートを受け取りますか？」みたいな項目を設けて、月に1回とかメールを送るようにするといいのでは。もしかしたらうっとしいかもしれないが。
- フォロワーから所属やら居住地を特定するというのは考えても居なかった。でもこれはどうやって気をつければいいのかのだろうか？
- 出す情報、出さない情報をふだん割りと意識して使っている方なので、解析結果はほぼ想定範囲内でした。逆に、普段意識していないような人には、ちょっと驚くような結果に見えるのかもしれませんが。そういう人達にとっては privasee.me は気づきを与えてくれるサービスかもしれないです。そういう意識してない層の人達に利用してもらうためのインセンティブ設計があるといいかと思いました。アイデアはないので言い逃げですが……。あとは、privasee.me も現状のように OAuth トークンを取得してデータを引っ張ってくるサードパーティサービスの形だと、privasee.me 自体も個人情報を取得してることになるので(仕方ないですが)、ぜひとも SNS ネイティブの機能として取り入れたいところですね。いい評価結果が出れば売りましょう
- もっとショッキングな結果(見た人が震えあがるような結果)が出れば面白かったのですが、今後の課題ですかね。誕生日の解析の仕方が(アイデア発勝負っぽくはありますが)面白いアプローチだと思ったので、こういった面白いアプローチを募集したりすると、とんでもない結果を解析できるものに仕上がるのでは、ということで、

一つの方法として提案します。

- 私は、基本的なプロフィールを非公開にしたいという考えがありません。SNS は実名で登録し、プロフィールを公開していますし、自分の個人情報を含む投稿も行っています。
- 以上が私の SNS 利用に関する考え方です。その下で、privasee.me に期待することを意見します。
- 私としては、第三者に知られているか気になるのは、名前や性別といった基本的なプロフィールではありません。自分の趣味嗜好や、アリバイ（参加したパーティの集合写真に自分が写ってしまっていて、遊んでたことがバレたりしますよね）、交友関係（twitter のリプライの飛ばし合いで浮気がバレたりしますよね）等です。privasee.me には、こういったトリッキーな解析を期待します。”
- ログインすると、” DoesNotExist at /top/” とでてトップ画面に進めないことがありました。解決策もわからず、戸惑いました。
- 分析項目を増やして、精度をさらにあげて欲しいと思います。
- 個人情報保護の観点で SNS 解析したのはおもしろい。何回も使ってもらうのは工夫が必要に思うので、繰り返し使うメリットが明確になるといいと思う
- そもそも、情報漏洩に対して意識が高かったです。そのために、僕の場合は、漏洩に対する意識向上には本システムは有効ではありませんでした。ただし、本当にどの程度漏洩しているのかを確かめ、自分の公開予定範囲と一致しているかを判定するのにとても有効でした。
- 今の形態で十分に素晴らしいサービスだと思いますが、結果表示をもっとグラフィカルにするとよりよいのではないのでしょうか？余力がありましたらご検討下さい。
- コンセプトを「プライバシー保護のため」としてらっしゃいますが、コンセプトは違えど似たような解析サイトはいくつかありますね。それらよりも精度がよくなれば、使おうという気になります。
- Googlechrome ではログインできなかった。また普段使っているように home に戻るリンクを使わず、ie の戻るボタンを使ったら 503 かなんかのエラーが出た。戻るボタンを使わんようにとか注意書きを書いてみては？
- とりあえず、1 回使って、結果をみて「へえ」って思っただけで終わる気がする。そこで意識して気をつけるようにする人はいると思うが、特に繰り返し使いたいとは思わないのではないかな。それから、その情報がなぜ危ないのか。どの程度危ないのかがわからない。
- 自分の個人情報が漏えいしてしまっている危険性を肌で感じることはできましたが、個人情報の取り扱い方についての関心は変化しませんでした。この解析結果が、実際にどの程度、個人情報の漏えいにつながる危険性があるのかが実感できなかったからだと思います。情報漏えいしている可能性と、それがどのくらい危険なことなのかを示す指標のようなものがあれば、ユーザに” 個人情報の漏えいに伴う危機感” を抱かせることができるのではないかと思います。※この意見が、Privasee.me をサービスとして提供するにあたっての目的から外れている場合は申し訳ありません。”
- 率直に、とても面白いなと思いました。私は好きです。
- プライバシーをチェックするサービスなので、サービス自体の信頼性やセキュリティも重要だと思います。より使われるようになるには、どういう風に解析しているのか、非公開情報にどれだけアクセスするのかといったことをちゃんと説明して、ユーザに安心を与えることが必要になるかなと思います。大きな企業と組んでそこから出すのも効果的かもしれません。
- あくまで私の場合ですが、解析器の精度があまり良くありませんでした。しかし、サー

バ側にいろんなデータがあると思いますので、それを用いデータマイニングを行うと、いろんな面白い、そして恐ろしい結果がでると思います。プライバシーを考えるととても良いウェブサービスだと思います。今後の解析器の精度向上および、Facebook のレスポンスに期待したいです。むちゃくちゃな文章になってもうしわけありませんでした。

質問14. その他

- 全体的に質問の意図が分かりづらい。SNS 別のアンケートを用意した方が良いと思う。
- 解析結果に関するアンケートに「完全に一致しなかった」という項目がありませんでした。代わりに「一致したもの・しないものが混じっていた」を選びました。性別判定の結果に関するアンケートが 2 回ありましたが（トップ画面の解析に対して/その他の解析に対して）、違いがわかりませんでした。
- facebook に関しては、一番会話している相手は出さないのでしょうか。性別の他に年齢は推測できませんか。今出ているもの以上に、見つかったらこまるような情報を発言していないか、解析できませんか。（例えば、飲酒運転や侮辱発言）
- 質問 7~9 で似たようなフレーズが並んだので多少混乱した。自由記述による回答を求める質問に関して、それらの質問文がアバウトであると感じた。
- アンケートの質問で意図がわからないのが幾つかあった。（たとえば、名前というのは本名なのかどうか等）アンケートで全く一致しないの選択肢がない。
- とても興味深いので、是非ここで終わらせず、改善を行ってパブリックサービスとして提供してほしいと思った。
- コンセプトは非常に興味深く、有用なものだと思います。解析精度や解析項目の詳細化が進めばより使いたいサービスになると思いました。さらに、解析結果からの個人情報類推等の新機能実装が行われれば、人気サービスになることも可能だと感じました。
- 最初の方にあった、見ず知らずの人の友人リクエストを承認するかどうかという設問は、Twitter かそれ以外かで大きく変わると思う。自分の場合、実名匿名関係なく、facebook と mixi では知ってる人のみリクエストを承認する。Twitter では、知ってる人だけでなく知らない人でも自分に価値のあることをつぶやいてたらフォローし返す。
- ここまでの自由意見のところに書いたので、特になしです。
- 自由記述が複数あるのですが、なんとなく設問が似てるので（特に質問 13 と 14）、たとえば「13 で書いちゃったけど 14 の設問文を読むと 14 に書いたほうが適切なんじゃないか？えーでも 13 に書いちゃったしいいか」って思うことがありました。
- アンケートについての要望ですが、SNS を利用しているうえで、知らない人からの友人申請が来た場合について、あてはまるものを選択してください。（複数回答可）*の選択肢がそうとうわかりづらい、MECE でないように思いました（全部にチェックを付けない場合もありうる?）。"
- 自由意見をどこの自由意見欄に書こうか迷いました。
- SNS には自分がどんな情報を垂れ流しにしているか意識していないユーザが多いと感じていたので、そういったユーザに啓蒙するためにも非常に意義深いサービスだと感じた。
- 今後の改良を期待します。がんばってください！
- 見落としていたのかもしれませんが、設問数がわからなくて結構長く感じました。
- ジョン、田中のあんちゃん&PBL チームメンバの皆様、お疲れ様でした！！筑波の先導 ICT コースの質の高さが伺えるサービスでした。最後の総まとめも頑張ってください（^^また 3 月に会いましょう！
- これからに期待
- 研究おつかれさまでした。こんなもの作れちゃうんですね><
- アルゴリズムはすごいのですが、サービスとしてはどうなのでしょう？
- その他の意見は特にありません。
- 開発お疲れさまでした。今年のテーマも面白そうでうらやましいです。
- 質問 13 に思ったこと書いてしまったので、そちらでお願いします。