ソフトウェア開発ドキュメント匿名化ツールの試作

江川翔太　岡野浩三　楠本真二
大阪大学　大学院情報科学研究科
{s-egawa,okano,kusumoto}@ist.osaka-u.ac.jp

要旨

ソフトウェア工学研究では，実プロジェクトで開発されたドキュメント，ソースコード等のソフトウェア開発ドキュメントを調査する，あるいは，得られた研究成果をそれらに適用・評価することが重要である．一方，ソフトウェア開発ドキュメントの多くは著作権や機密保持の点で公開することは難しい．更に，個別の共同研究においても，企業から大学へ成果物の提供を行う際にそれらの情報の隠蔽，つまり匿名化を行う必要がある．しかし，複数のドキュメントを人手で匿名化しなければならず，その作業にかかる人的コストの削減が必要である．本稿では，匿名化にかかるコストを削減し，ソフトウェア開発ドキュメントの研究利用の促進を目的として，ソフトウェア開発ドキュメントの匿名化を支援するツールを開発する．ツールでは対象ドキュメントに対する形態素解析・類語認識や利用者とのインタラクティブなやりとりを通じて，固有表現やコンテキスト情報（固有表現以外に間接的に固有表現を特定できるもの）の匿名化を支援する．匿名化ツールを実装し，ある実プロジェクトの成果物を対象に匿名化がどの程度効果的に行えるかどうかを評価した．その結果，匿名箇所に対して約 *8* 割程度の適合率と再現率となった．

1. はじめに

ソフトウェア工学は，品質の高いソフトウェアを低コストで期限通りに開発し，効率よく保守するための技術を扱う学問分野である．その一分野である実証的ソフトウェア工学は，実際の開発現場のデータからのアプローチ通じて，生産性や品質を高める研究である．例えば，実プロジェクトで開発された要求仕様書や設計仕様書，ソースコード，テストデータ，管理データ等のソフトウェア開発ドキュメントを調査する，あるいは，調査結果により得られた研究成果を実際の開発プロジェクトに適用・評価することが重要である．

一方，実プロジェクトのデータには，ユーザや開発ベンダの担当者の名前や連絡先のような個人情報や，守秘義務がある情報が含まれる．従って，企業から大学あるいは研究機関へ実プロジェクトのデータを提供する際は開発ドキュメント等に含まれる個人情報等を匿名化する必要がある．しかし，大量のドキュメントに含まれる情報は膨大で人手による匿名化は多くのコストがかかるという問題があり，匿名化を支援するツール等による匿名化コスト削減が求められる．医療関係書類を対象とした自動匿名化手法が提案されている [1] が，医療関係書類で匿名化すべき情報とソフトウェア開発ドキュメントで匿名化すべき情報が持つ特徴が異なるため，そのまま利用することは難しい．

本稿ではソフトウェア開発ドキュメントの匿名化の支援を目的として試作した匿名化支援ツールについて述べる．ツールでは対象ドキュメントに対する形態素解析・類語認識を行い，人名，地名，組織名といった固有表現を抽出し，別の記号に置換して匿名化を行う．また，利用者とのインタラクティブなやりとりを通じて，正しく抽出されなかった固有表現の修正，固有表現以外に間接的に固有表現を特定できるもの（コンテキスト情報）の匿名化を行う．匿名化ツールを試作し，ある実プロジェクトの成果物を対象に匿名化がどの程度効果的に行えるかどうかを評価した．その結果，ツールが自動で行う匿名化の場合，適合率が 0.83，再現率が 0.79 となった．また，使用者による短時間の補正で，適合率と再現率は共に 1.00 となった．

以降，2. では本研究で使用する諸用語と匿名化におけ

る課題について述べる．3. では匿名化手順と匿名化ツールについて説明する．4. では試作したツールの評価実験と結果の考察を述べる．5. では主な結果と今後の課題をまとめる．最後に，6. で謝辞を述べる．

2. 準備

ここでは本稿で使用する諸用語と匿名化における課題についてまとめる．

2.1. 匿名化

本研究において，匿名化とは連絡先情報の置換，固有表現の置換およびコンテキスト情報の置換を指す．以下で連絡先情報の置換，固有表現の置換およびコンテキスト情報の置換の詳細を述べる．

2.1.1 連絡先情報の置換

連絡先情報は，メールアドレス，電話番号，郵便番号，住所の丁番地および URL を指す．ドキュメント中に出現するこれらの情報を，記号で置換する事で匿名化を行う．

2.1.2 固有表現の置換

本稿において，固有表現とは IREX[2] によって規定された 8 種類の分類の内，人名，地名および組織名を指す．ただし，人名では苗字と名前を区別する．ドキュメント中に出現するこれらの文字列を記号に置換する事で匿名化を行う．この時，固有表現の分類情報を保持する．また，ドキュメント中に同一の固有表現が複数出現する場合，それらを同一の記号で置換する．

2.1.3 コンテキスト情報の置換

ある文字列をもとに固有表現を推測できる場合，そのような文字列をコンテキスト情報と定義し匿名化の対象とする．例として，銀行名，銀行の支店数および従業員数を含む銀行向け勘定系システム開発ドキュメントの匿名化を考える．固有表現の匿名化では銀行名の匿名化は成されるが，支店数と従業員数の匿名化は成されない．ここで，ドキュメントの読み手が意図的に各銀行の支店数と従業員数を調査すると銀行名が特定されてしまう恐れがある．この場合，コンテキスト情報は支店数や従業員数となる．これを防ぐためコンテキスト情報を匿名化する必要がある．

2.2. ソフトウェア開発ドキュメント匿名化における課題

2.2.1 コンテキスト情報の選定

コンテキスト情報は，開発されるソフトウェアの種類，守秘義務情報，顧客の希望，機能内容等，様々な観点で決まる．例えば，大学を対象とした履修登録システム開発ドキュメントの場合，納品先の大学名が匿名化の対象となるのであれば，コンテキスト情報は学部名や総学生数等が考えられる．また，銀行を対象とした勘定系システム開発ドキュメントの場合，コンテキスト情報は支店数や従業員数等が考えられる．従って，表記が同じ情報であっても，コンテキスト情報となる場合とならない場合がある．更に，匿名化されたドキュメントの利用目的によってはこのような数値情報が必要となる場合もある．

2.2.2 固有表現の分類

固有表現は同一の表記で複数のものを指す場合がある．例えば「大津」という人名には同名の地名が存在する．また，一般名詞と判別がつかない固有表現も存在する．例えば,「森」は一般名詞であるが，文脈により人名を指す場合がある．これらの文字列は形態素解析の際に分類を誤る恐れがある．

2.2.3 表記ゆれ

表記ゆれとは同じ意味の語句について異なる文字表記が付されることである．人名では漢字表記と読み仮名表記，組織名では略称や別称といった表記ゆれが存在する．例えば，大阪大学という組織には阪大や Osaka University 等の異なる表記が存在する．同一ドキュメント中に出現するこれらの表現を異なる記号で置換した場合，単一の組織を指しているにも関わらず，読み手に複数の組織として認識される恐れがある．そのため異なる表記を同一の記号で置換する必要がある．

以上述べた 3 つの課題に対して，本稿では,「コンテキスト情報の選定」と「固有表現の分類」はツールの利用者とのインタラクティブなやりとりにより対応し,「表記

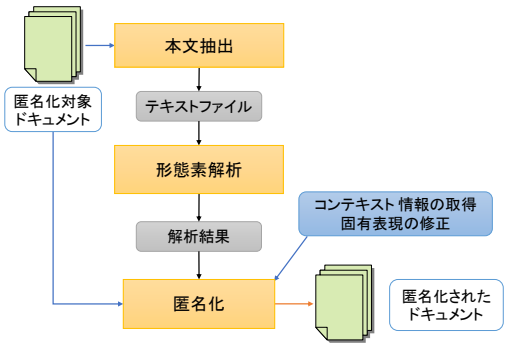

図 1. 匿名化手法

対象文：大阪大学の教務システムを開発する

形態素解析結果

大阪大学	名詞，固有名詞，組織，大阪大学，オオサカダイガク
の	助詞，連体化，の，ノ
教務	名詞，一般，教務，キョウム
システム	名詞，一般，システム，システム
を	助詞，格助詞，一般，を，ヲ
開発	名詞，サ変接続，開発，カイハツ
する	動詞，自立，サ変・スル，基本形，する，スル

図 2. 形態素解析結果の例

ゆれ」については，類語辞典を参照し同一の記号で置換する方法をとる．

3. 提案する匿名化手法

本章では，ソフトウェア開発ドキュメントの匿名化手法とその実装方法について説明する．

3.1. 概要

図 1 に匿名化手法を示す．手法は 3 つのステップから成る．まず，匿名化対象ドキュメントから本文を抽出しテキストファイルに変換する．次に，抽出した本文に対し形態素解析を行い本文を品詞ごとに分解する．そして，形態素解析によって固有名詞に分類された文字列に注目し，ドキュメント中に出現する固有表現を記号に置換する．この時，誤って分類された固有表現を正確に分類する．また，コンテキスト情報を取得する．以上で述べた 3 つのステップでドキュメントを匿名化する．今回試作した匿名化ツールでの具体的な内容については，3.2 以降で，実装内容とともに説明する．

3.2. 実装

匿名化手法の実装にあたっては，本文抽出部分を本文抽出プログラムとして開発し，形態素解析部分に既存の形態素解析ツールを利用した．また，匿名化部分を匿名化プログラムとして実装した．以降，各部分の詳細な実装について述べる．

なお，本試作にあたって，入力は Word と Excel 形式のファイル，出力は匿名化された Word と Excel 形式のファイルとする．

3.2.1 本文抽出プログラム

Apache POI ライブラリ [3] を用いて Word および Excel 形式のファイル本文を抽出し，形態素解析用のテキストファイルへ出力する．

3.2.2 形態素解析ツール

形態素解析ツールとして Cabocha[4] を用いた．Cabocha は Support Vector Machines に基づく日本語係り受け解析器で，高い解析精度を持つ．

Cabocha は外部プログラムであるため，ProcessBuilder を用いて実行する．このとき，Cabocha への入力として本文出力プログラムが出力したテキストファイルを与える．Cabocha は形態素解析結果が記述されたテキストファイルを出力する．Cabocha による形態素解析結果の例を図 2 に示す．

3.2.3 匿名化プログラム

匿名化プログラムは，固有表現の抽出，固有表現の修正，表記ゆれの取得，コンテキスト情報の取得および匿名化対象情報の置換という 5 つの動作を行う．以降，各動作の実装について説明する．

表 1. 連絡先情報の正規表現と置換後の文字列

連絡先情報	正規表現	置換後の文字列
メールアドレス	[\w\.\-]+@(?:[\w\-]+\.)+[\w\-]+	xxxx@xxxx
電話番号	\d{1,4}?-\d{1,4}?-\d{1,4}	xxxx-xxxx-xxxx
郵便番号	\d{3}-\d{4}	xxx-xxxx
住所の丁番地	((丁目|番地?|号)?[-ー－−]?)$	xxx-xxx-xxx
URL	(https?|ftp)(:\/\/[-_.! *\'()a-zA-Z0-9;\/?:\@\&=+$,%#]+)	URL

固有表現の抽出では，まず，固有表現を分類するために ArrayList を用いて，苗字リスト，名前リスト，地名リストおよび組織名リストを作成する．次に，形態素解析ツールの解析結果を 1 行ずつ読み込み「固有名詞」と記述された行の先頭の文字列を固有表現として抽出する．そして，図 2 で示されるような，同一行に記述された固有表現の分類を参照し各リストに記憶する．

固有表現の修正では，まず，抽出した固有表現をリストごとにユーザに提示する．次に，それぞれのリストについて固有表現の追加と削除をユーザへ依頼する．

表記ゆれの取得では，類語辞典として Weblio 類語辞典 [5] を用いる．Weblio 類語辞典において検索結果ページの URL は，“http://thesaurus.weblio.jp/content/検索ワードの URL エンコード結果” となっている．まず，リストに保存した組織名と同数の表記ゆれリストを作成する．次に，組織名の URL エンコードを行い検索結果ページの URL を取得する．そして，取得した URL を用いて検索結果ページにアクセスし HTML を取得する．最後に，表記ゆれが記述された部分を HTML から抽出し表記ゆれをリストに記憶する．

コンテキスト情報の取得では，まず，コンテキスト情報を記憶するためのリストを作成する．次に，コンテキスト情報の入力をユーザへ依頼し，ユーザが入力した文字列をリストへ追加する．

連絡先情報の置換では，まず，ドキュメントから本文を 1 行ずつ抽出する．そして，正規表現によるパターンマッチを行い，マッチした場合は連絡先情報を別の記号に置換する．連絡先情報の正規表現と置換後の文字列を表 1 に示す．

固有表現の置換では，まず，ドキュメントから本文を 1 行ずつ抽出する．そして，各固有表現リストを参照し，抽出した文に固有表現が含まれている場合は別の文字列に置換する．各リストに含まれる固有表現の置換後の文字列を表 2 に示す．x はリストに記憶された順番を表す．

表 2. 固有表現の置換後の文字列

リスト名	置換後の文字列
組織名リスト	組織名 x
苗字リスト	人名（姓）x
名前リスト	人名（名）x
地名リスト	地名 x

コンテキスト情報の置換では，まず，ドキュメントから本文を 1 行ずつ抽出する．そして，コンテキスト情報リストを参照し，抽出した文にコンテキスト情報が含まれている場合は「その他 x」という文字列に置換する．x はリストに記憶された順番を表す．

3.3. 適用例

本節では図 3 に示すツールの適用対象となるドキュメントの一部を用いて匿名化を説明する．

本文抽出プログラムは，図 3 に示すドキュメントから本文を抽出しテキストファイルに変換する．

形態素解析ツールは，本文を抽出したテキストファイルに対し形態素解析を行う．

解析結果から固有表現を抽出すると，「江川」は苗字，「吹田」，「和歌山」および「栄谷」は地名，「和歌山大学」は組織名として分類される．しかし，本来人名として扱われるべき「吹田」という固有表現が地名として分類されている．そのため，ユーザに固有表現の修正を依頼し，「吹田」を人名として再分類する．

固有表現を修正した後，Weblio 類語辞典にアクセスし，組織名の表記ゆれを取得する．和歌山大学の略称および別称として，「Wakayama University」，「和大」，「和歌山大学付属学校」および「わかやまだいがく」が提示されるため，これらの文字列を表記ゆれとして取得する．その後，ユーザに入力を依頼し，コンテキスト情報を取

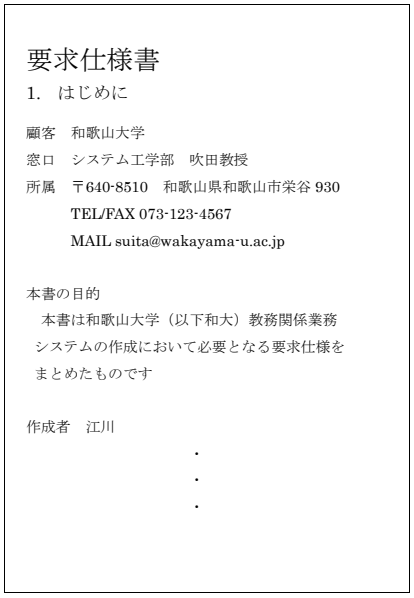
要求仕様書
1. はじめに
顧客　和歌山大学
窓口　システム工学部　吹田教授
所属　〒640-8510　和歌山県和歌山市栄谷 930
TEL/FAX 073-123-4567
MAIL suita@wakayama-u.ac.jp
本書の目的
本書は和歌山大学（以下和大）教務関係業務
システムの作成において必要となる要求仕様を
まとめたものです
作成者　江川

図 3. 匿名化対象ドキュメントの一部

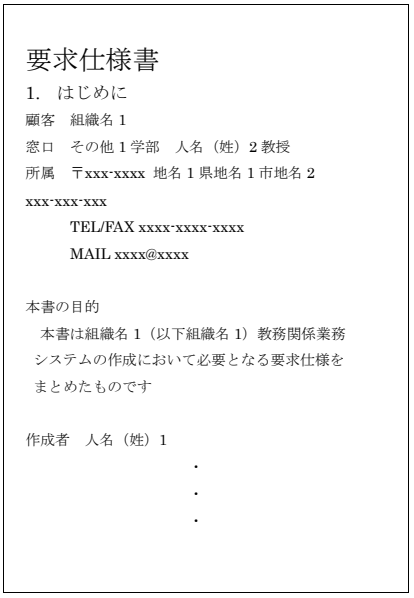
要求仕様書
1. はじめに
顧客　組織名 1
窓口　その他 1 学部　人名（姓）2 教授
所属　〒xxx-xxxx 地名 1 県地名 1 市地名 2
xxx-xxx-xxx
TEL/FAX xxxx-xxxx-xxxx
MAIL xxxx@xxxx
本書の目的
本書は組織名 1（以下組織名 1）教務関係業務
システムの作成において必要となる要求仕様を
まとめたものです
作成者　人名（姓）1

図 4. 匿名化されたドキュメントの一部

得する．そして，連絡先情報，固有表現およびコンテキスト情報の置換を行うことでドキュメントを匿名化する．匿名化されたドキュメントの一部を図 4 に示す．

4. 評価実験

ここでは試作した匿名化ツールを実プロジェクトのソフトウェアドキュメントに適用し，有効性の評価と考察を行う．

4.1. 実験対象

ツールの適用対象とするドキュメントに IT Spiral 実プロジェクト教材 [6] を用いる．これは，和歌山大学を対象とした履修登録システム開発の際に得られた要求定義書やプロジェクト管理書，画面設計書等のドキュメントが教材として大学に提供されたものである．本実験では，特定の大学を対象としたシステム開発であることが推測可能となる，学部名やコース名をコンテキスト情報として扱う．ドキュメントは全 97 ファイル存在し，ファイルサイズは合計 16.9MB，ドキュメント内に出現する固有表現数は 170，コンテキスト情報数は 21 となる．

表 3. 実験結果

固有表現数	ツール出力	出力正解数	適合率	再現率
170	162	134	0.83	0.79

4.2. 実験手順

実験対象ドキュメントに対して匿名化ツールを適用した（利用者による修正無し）．その結果から，ツールが匿名化した文字列数と正確に匿名化した固有表現数をカウントし適合率と再現率を求めた．実験における適合率はツールがどれだけ正確に匿名化を行ったかを示す指標であり，以下のように定義する．

$$\text{適合率} = \frac{\text{ツールが正確に匿名化した固有表現数}}{\text{ツールが匿名化した文字列数}}$$

実験における再現率はツールがどれだけ情報を漏らさずに匿名化を行ったかを示す指標であり，以下のように定義する．

$$\text{再現率} = \frac{\text{ツールが正確に匿名化した固有表現数}}{\text{ドキュメント内に出現する固有表現数}}$$

4.3. 実験結果

実験結果を表 3 に示す．表の横軸の内，“固有表現数”はドキュメント中に出現する固有表現数，“ツール出力”

はツールが匿名化した文字列数，“出力正解数” はツールが正確に匿名化した固有表現数，“適合率” はツールが匿名化した文字列の内匿名化すべき固有表現数の割合，“再現率” はドキュメント内に出現する固有表現数の内ツールが匿名化した固有表現数の割合を表す．

ツールの実行時間は約 17 秒であり，また，使用者が固有表現の修正やコンテキスト情報の追加を 10 分程度行うことで適合率と再現率は共に 1.00 となった．

4.4. 考察

ツールが行った匿名化の内，正確に匿名化が行われなかった例について述べる．

適合率は 0.83 となっている．適合率はツールが誤った匿名化を行った場合に低下する．ツールが誤って匿名化した文字列として,「大津」という人名が挙げられる.「大津」という人名には，同表記の地名が存在するため文脈により誤って地名と分類される場合が存在した．

また，再現率は 0.79 となっている．再現率はツールが匿名化すべき固有表現を匿名化できなかった場合に低下する．ツールが匿名化できなかった固有表現として，人名の読み仮名と「日本システム技術株式会社」という一般名詞が組み合わさった組織名が挙げられる．人名の読み仮名が固有表現として抽出されないのは，複数の一般名詞が組み合わさった文字列として扱われるためである．

以上のような誤った匿名化が行われないようにするためには，頻出する固有名詞を含めた辞書をあらかじめ用意する必要がある．例えば，開発に関わる人物や組織名のリストを作成しツールがそれを参照すれば，実験における誤りは全て正確に匿名化されると考えられる．また，使用者によって頻繁に修正が行われる組織名や人名を学習することで，誤った匿名化を防ぐことが可能になると考えられる．

5. おわりに

本稿では，ソフトウェア開発ドキュメントの匿名化を支援するツールを開発しその性能を評価した．その結果，ツールが自動で行う匿名化の場合，適合率が 0.83，再現率が 0.79 となった．また，使用者による短時間の補正で，適合率と再現率は共に 1.00 となった．

今回試作したツールでは，コンテキスト情報の匿名化について，ほとんど利用者が指定する方式になっている．一度利用したコンテキスト情報の学習やソフトウェアの対象ドメインに応じて，コンテキスト情報例を提示する等，利用者を支援する機能を実現する必要がある．また，対応ファイル形式の拡張やツールのインターフェース面の改良を行うことも必要である．更に，実際の現場で作成された多くのソフトウェア開発ドキュメントにツールを適用することで，有用性の評価を行うことも今後の課題の一つである．

6. 謝辞

本研究を行うにあたり貴重なコメントをいただいた株式会社エヌ・ティ・ティ・データの端山毅氏，山本英之氏，中村英恵氏に感謝いたします．

参考文献

[1] Stephane Meystre, F Friedlin, Brett South, Shuying Shen, and Matthew Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, Vol. 10, No. 1, p. 70, 2010.

[2] Satoshi Sekine and Hitoshi Isahara. IREX: IR & IE Evaluation Project in Japanese. In *LREC*, pp. 1977–1980, 2000.

[3] Apache poi - the java api for microsoft documents. *http://poi.apache.org/*.

[4] K Taku. Cabocha: Yet another japanese dependency structure analyzer. Technical report, Technical report, Nara Institute of Science and Technology, 2004.

[5] 類語辞典・シソーラス - weblio 辞書. *http://thesaurus.weblio.jp/*.

[6] 実プロジェクト教材. *http://it-spiral.ist.osaka-u.ac.jp/project/education.html*.