1E3-3

インシデントレポートの自動分類とその分析環境の構築と評価

Labeling of Incident Reports and Development and Evaluation of Analysis Environments

稗方和夫[*1] 大和裕幸[*1] 中村覚[*1] 岡田伊策[*2] 齋藤稔[*2] 安藤峻[*3]
Kazuo HIEKATA Hiroyuki YAMATO Satoru NAKAMURA Isaac OKADA Minoru SAITO Takashi ANDO

[*1] 東京大学大学院新領域創成科学研究科
Graduate School of Frontier Sciences, THE UNIVERSITY OF TOKYO

[*2] 富士通株式会社 SI 技術サポート本部
SYSTEM INTEGRATION TECHNOLOGY SUPPORT UNIT, FUJITSU LIMITED

[*3] 株式会社ウニークス
UNICUS Co., Ltd

An enterprise involved System Information accumulates knowledge about incidents which occurred under clients' use as incident reports. This paper aims to reuse knowledge of existing incident reports in order to solve new incidents efficiently and the program to label those reports by use of machine learning is developed. In addition, the platform which handles common processes in analysis such as accumulating resources and saving those metadata is proposed. The programs for specific processes which depend on texts run as plug-ins on this platform. Developed program to label reports in this paper also runs on proposed platform as a plug-in, and the accuracy of the programs is verified by actual incident reports in an enterprise.

1. はじめに

1.1 背景

情報システム企業 A 社では情報システム製品の顧客環境での運用時等のインシデント情報をインシデントレポート(以下、レポート)として蓄積し、新規のインシデントが発生した際に過去の類似インシデントに関するレポートを参照することでインシデントの解決に役立てている。

現在の問題点として、既存レポートを参照する際には全文検索が用いられているが、蓄積された膨大な数のレポートを十分に絞り込めないケースが存在する。そこで知識抽出という観点から、レポートをカテゴリ毎に分類することで検索能力を向上させ、目的とするレポートへのアクセスを容易にする試みがなされている。しかし現在は人手でレポートがカテゴリに分類されているため、分類コストとカテゴリの分類基準が曖昧である。そのためレポートが分類しきれていない問題とカテゴリへの分類がぶれる問題により、結果としてカテゴリを用いた検索が有効に機能していない。

著者らはこの問題に対して、これまでに文書の類似度を利用したカテゴリ付与によりこの問題の解決への取り組みを行ってきたが[稗方 2013]、本報では整備した研究プラットホームの評価を行う。

1.2 目的

本研究では自然言語処理技術、機械学習技術を用いて蓄積されたレポートをカテゴリに自動分類することを目的とする。また分類対象に依存したテキスト処理以外の共通部分をプラットホーム化することで横展開可能な文書自動分類プログラムを提案する。さらに情報システム企業 A 社のデータに対して、本プログラム上でカテゴリの自動付与を行い、その有用性の検証を行う。

2. インシデントレポート

2.1 インシデントレポート

レポートは図 1 に示すように「OS」「製品名」「質問概要」「回答概要」等の複数の項目によって構造化され記述されている。「質問概要」項目には発生したインシデントの症状が記述されており、どのような処置を行った際にどのような事象が発生したかが要約して記述されている。「回答概要」項目にはインシデントの原因と解決策が記述されており、過去のレポートを用いて新規のインシデントを解決する際には、主にこの項目に記述された情報が参照される。

なお本研究では図 1 に示した項目が XML 形式で構造化されたデータを使用する。

ID:a1234-5678 **日付:**2013/1/1 **OS:** Windows **製品:**製品B

1. 質問概要
製品Aの起動シェルを置き換え後、OS再起動を実施しましたが製品Bが起動しなくなりました。

2. 回答概要
Dコマンドを実行していないのが原因です。
マシンブート時の製品Bの自動起動設定がされているか確認してください。

3. ヒアリング
・OS再起動の時間 ・システム等の変更点はないか ・復旧(製品A起動)方法

4. 原因要約
Dコマンドを実行していないのが原因です。

5. 処理要約
マシンブート時の製品Bの自動起動設定がされているか確認してください。

6. 参考情報
・マニュアル 製品B運用ガイド
付録C製品B統合コマンドによる運用操作
＞C.3 製品Bの起動
C.3.5 マシンブート時の製品Bの自動起動

・過去事例 a3456-9876

図 1 インシデントレポートの例

2.2 分類カテゴリ

情報システム企業 A 社はレポートをカテゴリ毎に分類し、検索能力を向上させることによって目的とするレポートへのアクセスを容易にする試みを行っている。本研究で対象とするアプリケーションサーバ A に関するレポートについて、その症状で分類した 11 カテゴリを図 2 に示す。

図 2　インシデントの症状に関する分類

なお本研究ではレポートをカテゴリに分類することを、各レポートに対して図 2 に示すカテゴリから該当するすべてのカテゴリを付与するラベル付け問題へと帰着させ論を進める。

2.3 カテゴリを用いた検索方法の有用性の検証

ここで 2.2 章において説明したカテゴリを用いた検索方式（以下、カテゴリ検索）の有用性の検証を行う。

ここでは「ソフトウェア B が停止した」というインシデントが新規に発生したと仮定し、本インシデントの解決に寄与するレポート（以下、正解レポート）を過去インシデントレポートから全文検索とカテゴリ検索によって検索し、両検索方法の検索性能の比較を行う。なお本実験を行うにあたり、検索対象とする 799 件のレポートから正解レポートを予め抽出した。また被験者として業務経験の少ない被験者 1 名を選択した。

具体的な検索の流れを説明する。検索者には発生したインシデントの事象に関する情報（本実験では「ソフトウェア B が停止した」）が予め与えられる。この情報を用いて、全文検索では検索者が自ら検索クエリを決定し検索する。一方カテゴリ検索においては、図 2 に示したカテゴリと照合し検索する。

全文検索とカテゴリ検索の検索結果を表 1 に示す。

表 1　全文検索とカテゴリ検索の検索結果の比較

全文検索クエリ		検索結果数	正解レポート数	検索結果に含まれる正解レポート数
ソフトウェアB		187	17	17
異常終了		69		15
停止		88		16
ダウン		21		0
ソフトウェアB	異常終了	32		15
ソフトウェアB	停止	38		16

選択したカテゴリ	検索結果数	正解レポート数	検索結果に含まれる正解レポート数
ソフトウェアBが停止できない	14	17	14

表 1 上部に全文検索による検索結果、下部にはカテゴリ検索による検索結果を示す。またそれぞれ左から検索に使用したクエリ、検索結果数、正解レポート数、検索結果に含まれる正解レポート数を示す。

全文検索では「停止」「異常終了」「ダウン」等の検索クエリが複数回入力されているが、検索結果数に対する正解レポート数の割合が小さくなっている。一方カテゴリ検索では、クエリとして選択したカテゴリに分類されたレポート全てが正解レポートであり、検索結果数に対する正解レポート数が高くなっている。この結果から、カテゴリ検索の導入は検索能力の向上に寄与することが確認できる。

3. 分析プラットホーム「KASHIWADE」

3.1 概要

本研究では分類対象に依存したテキスト処理以外の共通部分をプラットホームとして提供する「KASHIWADE」（以下、KASHIWADE）を開発した。その概要を図 3 に示す。このプラットホームは Web ブラウザから使用する Web アプリケーションとして構築されている。次節より KASHIWADE が保有する各機能について説明する。

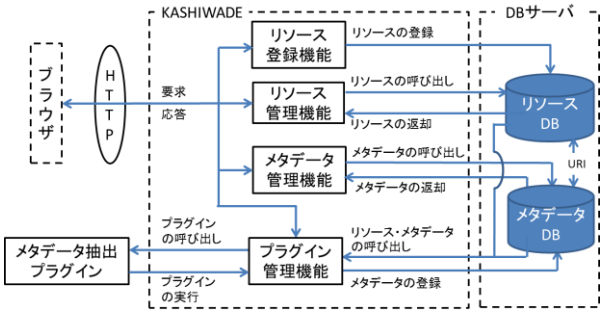

図 3　KASHIWADE 概要

3.2 リソース管理画面

リソースは文書登録 UI を通じて KASHIWADE に登録される。リソースが持つ情報は MySQL によって構築されたリソース DB とメタデータ DB に保存される。リソース DB にはバイナリデータと URI が保存され、メタデータ DB にはメタデータとしてリソース名やグループ名、URI が保存される。これら DB は URI によって結合される。

登録されたリソースはリソース管理画面に表示され、メタデータ検索によるリソースの絞り込みや、各リソースのダウンロードや更新、削除を行うことができる。

3.3 メタデータ管理機能

特定のリソースについてのメタデータフィールドとメタデータバリューは、図 4 に示すメタデータ管理画面に表示され、メタデータバリューの参照や更新を行うことができる。

図 4　メタデータ管理画面

3.4 メタデータ抽出プラグイン管理機能

本プラットホームはプラグインを読み込み、実行する機能を備えている。プラグインはプラットホームに保存されたバイナリデータと、リソースに付与されたメタデータ(フィールドとバリューのセット)を読み込み、既存メタデータフィールドのバリューの更新や新規メタデータの追加を行う。

プラグインは図 5 に示す管理プラグイン画面に一覧表示され、選択したメタデータフィールドのバリュー別、リソースの拡張子別にプラグインを実行できる。

図 5　プラグイン管理画面

4. ラベル自動付与プログラム

4.1 KASHIWADE を利用した前処理

KASHIWADE を利用し、前処理として分析対象に依存しないテキスト処理を行う。その流れを図 6 に示す。

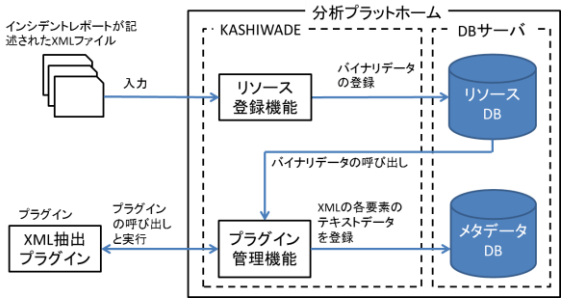

図 6　KASHIWADE を利用した前処理

まずレポートが記述された XML ファイルを訓練データとして KASHIWADE に登録することで、バイナリデータがリソース DB に、レポート名がメタデータ DB に保存される。次に XML から各要素を抽出する XML 処理プログラムをプラグインとして実行することで、2.1 章で説明した各項目がメタデータフィールドとして、そこに記述されたテキストがメタデータバリューとしてメタデータ DB に保存される。さらに人手によって付与されたラベル(以下、正解ラベル)もメタデータとして保存する。この前処理によって次節以降で必要となるテキストデータや正解ラベルが KASHIWADE に保存され、これらを入力データとして用いることで分析対象に依存したテキスト処理プラグインを実行できる。

4.2 特徴ベクトルの作成

ここではレポートに記述されたテキストから特徴ベクトルを作成する方法について説明する。ここで特徴ベクトルとはテキストについて形態素解析を行い、分割された形態素に関する出現頻度を特徴量として持つベクトルのことである。なお、形態素解析エンジンには MeCab を使用した。

特徴量として使用する形態素を選定する際、平らの研究[平 2005]、中川らの研究[中川　2003]を参考とした。品詞として「一般名詞」「固有名詞」「未定義語」「サ変接続名詞」を持つ形態素を抽出し、また「ない」「ん」のような否定語も特徴量として抽出した。これは「A が停止する」「A が停止しない」のように否定語の有無によってインシデントの症状の意味内容が反転するためである。また専門用語に関して「アン/インストール」「ジョブ/ネット」(「/」は一般的な形態素解析器での分割単位)等の用語を IT 用語集から約 3000 件登録した。さらに「管理/コンソール」「異常/終了」等の業務分野に特化した専門用語については、出現頻度の高い複合名詞から手動で抽出することで登録した。

ここまでの処理によって作成された特徴ベクトルについて、さらに全特徴量の合計頻度で除すことによる正規化と、tf-idf 法による特徴量の重み付けを行う。

4.3 ラベル自動付与プログラムの生成

4.2 章で作成した特徴ベクトルを用いて、ラベル自動付与プログラムを生成する。筆者らはこれまで線形カーネルを利用したソフトマージン SVM を用いて、インシデントレポート毎に図 2 で示した各ラベルが付与されるか否かの二値を判別する識別器を生成し、ラベルを自動付与するプログラムを提案した。しかし SVM による付与結果について、後述する各ラベルが付与されたレポートに関する重心ベクトルとの類似度を用いた付与結果の精度と大きな違いがなかったため、本研究では計算量の少ない後者によるプログラムを新たに提案する。

まずラベルが付与されている訓練データから各ラベルの重心ベクトルを生成する。次にこれらの重心ベクトルと、ラベルが未付与のレポートから作成した特徴ベクトルとの類似度をコサイン類似度によって算出する。これによりラベルが未付与のレポートに対して、各ラベルとの類似度が定量的にソートされ、付与する最大ラベル数と類似度に関する閾値の条件を満たすラベルが未付与のレポートに付与される。

5. 実験

5.1 実験概要

本研究では情報システム企業 A 社のレポートについて、人手によって正解ラベルが付与されたレポート 799 件を対象として実験を行った。なお、ここでは 2.2 章で挙げたカテゴリに関して 799 件のレポートに付与された 10 カテゴリを付与対象とする。

評価は交叉検定による Precision と Recall によって行う。Precision とは識別器によって付与されたラベルの中に正解ラベルが含まれる割合、Recall は正解ラベルの中に付与された識別器によって付与されたラベルが含まれる割合を示す指標である。訓練データを一件のテストデータと残りの訓練データに分割し、テストデータの正解ラベルと識別器によって付与されたラベルを比較することを全訓練データに対して行い、それらの Precision と Recall の平均値を結果として出力する。

また評価は 799 件から無作為に抽出した 100 件について、人手によって再度ラベルが付与され直したラベルを正解ラベルとして使用した。これは 799 件のレポートには付与されるべきラベルがすべて付与されていないというヒアリング結果に基づく。

また本実験ではレポートの記述された項目のうち、「質問概要」項目に記述された内容を用いて特徴ベクトルを作成した。こ

れは現場において「質問概要」項目に記述された内容に基づいてラベルの付与を行っているというヒアリング結果に基づく。

5.2 実験結果

図 7 に実験結果を示す。評価に使用した 100 件のレポートには平均 1.79 個のラベルが付与されていたため、4.3 章で説明したラベル自動付与プログラムにおいて、各レポートに対して類似度が高い上位 2 件のラベルを付与した。さらに類似度に関して閾値を 0 から 0.5 まで変更して評価を行った。

結果として閾値を設けない場合が Precision、Recall 共に最大値となり、Precision が 57.5%、Recall が 67.2%を示した。

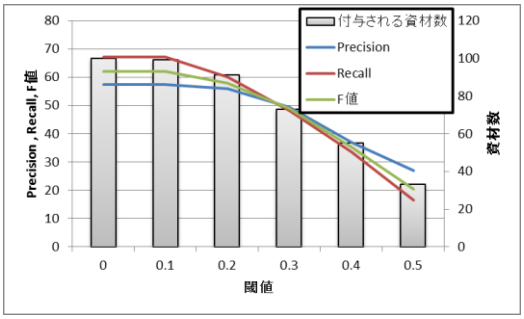

図 7　ラベル付与結果

6. 考察

6.1 ラベルの付与結果に関する考察

ここでは 5.2 章で得た付与結果の考察を行う。図 8 にラベル別の付与結果を示す。

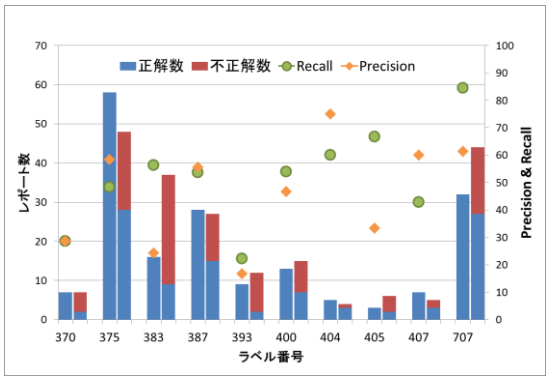

図 8　ラベル別の付与結果

Recall が高いラベル「707:運用中にログメッセージが出力された」については本ラベルが付与されるレポートの多くが「エラーメッセージ」という用語を含み、このキーワードが本ラベルを特徴づけているために高い Recall を示している。逆に Precision が低い理由としては、多くのインシデントはエラーメッセージの出力とともに発生するため、他のラベルが付与されていたレポートにも「エラーメッセージ」という用語が多く出現することで本ラベルが付与され、結果として Precision が低い値を示している。この件について「エラーメッセージが記述されているインシデントに関しては過去のレポートを参照せずに製品マニュアルを検索する」というヒアリング結果も得られているため、本ラベルを既存の分類カテゴリから削除する等を検討する必要がある。

Recall が低いラベル「393:操作できない」にでは、「操作できない」という症状を表現するための固有の表現が存在せず、結果として本ラベルを付与するための特徴量が抽出できていないため、Recall が低い値を示している。この問題については形態素の出現頻度を用いた特徴量の抽出方法では限界であり、オントロジーを用いた表記揺れの吸収等を行う必要がある。

6.2 分析プラットホームに関する考察と評価

本研究では分析対象に依存したテキスト処理以外の共通部分をプラットホーム化した KASHIWADE を提案した。リソースやメタデータの管理はプラットホーム上で行い、分析対象に依存した処理をプラグインとして実行する。これにより、研究テーマごとに個別にリソースを管理する環境を開発する必要がなく、研究者はアルゴリズムの実装に集中することができる。またある研究テーマで実装したプログラムが過去の研究テーマで作成したメタデータを再利用することもできる。

この考察を確認するために追加実験を行う。ここでは本研究で対象としたレポート 799 件に対して、インシデントの「発生した原因」が記述された「回答概要」項目を分析対象とし、K-means 法によるクラスタリングを行った。なお、ここで分析対象として用いた「回答概要」項目のテキスト内容は 4.1 章の XML 抽出プラグインによって KASHIWADE 上に登録済みのメタデータである。

クラスタ数を 10 として K-means 法を実装したプラグインを実行した結果、全レポートの約 1 割にあたる 66 件のレポートが、「メモリ」「ヒープ」「不足」等をキーワードとして持つクラスタに分類された。これは”メモリのヒープ領域の不足”によって生じるインシデントが一定数存在することを示し、既存の”インシデントの症状に関する分類”の他に、インシデントの発生原因によっても分類できる可能性を示唆している。

このように KASHIWADE 上で研究を行うことにより、過去のメタデータやプラグインの再利用や分析目的に依存したアルゴリズムの実装に注力することが可能となり、研究者の労力低減に寄与できることが確認できた。

7. 結論

本研究では、レポートに対するラベルの自動付与手法を提案した。情報システム企業 A 社のレポート 799 件に対してラベルの付与結果を交叉検定によって評価した結果、Precision が 6 割、Recall が 7 割という値を示した。今後は実利用によるログ等を用いて、付与したラベルの有用性の評価を行う。

また本研究では分類対象に依存したテキスト処理以外の共通部分をプラットホーム化した KASHIWADE を開発・提案し、分析に用いるプログラムをプラグインとして実行することで、分析における労力の低減などの有用性の評価を行った。

謝辞

本研究を行うにあたり、多大なご指導をいただいた富士通株式会社ミドルウェア事業本部の方々に感謝いたします。

参考文献

[稗方　2013] 稗方和夫, 大和裕幸, 中村覚, 岡田伊策, 齋藤稔, 安藤峻: インシデントレポートの自動分類とその分析環境の構築, 人工知能学会第 18 回知識・技術・技能の伝承支援研究会(SIG-KST), 2013.

[平　2005] 平博順, 春野雅彦: Support Vector Machine によるテキスト分類における属性選択, 情報処理学会論文誌, 情報処理学会, 2005.

[中川　2003] 中川裕志, 湯本紘彰, 森辰則: 出現頻度と連接頻度に基づく専門用語抽出, 自然言語処理, 言語処理学会, 2003.