Vol.2017-SE-195 No.22
2017/3/12

文書中の単語出現頻度を利用したトピックモデル洗練化

高橋 仁[1,a)]　中川 博之[1,b)]　土屋 達弘[1,c)]

概要：近年では、開発環境の変化に伴って開発者が大量の自然言語文書を扱う機会が増えており，文書をトピック分類するためのトピックモデルである LDA が注目されている．LDA の適用においては，前処理で用いられるストップワードリストによって一般語をフィルタリングし，より正確なトピック分類を試みるが，通常のストップワードリストでは対象文書にのみ頻出する単語に対応できないという問題があった．また，1 トピックに集約されるべき文書が複数トピックに分散してしまう問題があった．本研究では，これらの問題を解消するため，LDA 適用の前後に対象文書からのストップワード抽出と類似トピック統合の 2 つの処理を追加する手法を提案する．この提案手法では，対象となる自然言語文書から頻出語を特定しストップワードリストに加えるべき単語を抽出することで，対象文書から適切なストップワードリストを作成する．また，分類されたトピックについて構成する単語の類似度からそれぞれのトピック間距離を算出し類似トピックを統合する．提案手法を 3 種類の文書に適用する実験を行った結果，通常のトピック分類よりも正確性が向上していることが確認できた．

1. はじめに

近年では，開発者がテストケース記述やフィードバックコメントなどの自然言語で書かれた文書を開発に用いる機会が増えている．自然言語で記述された文書は開発者にとって有益な情報を含んでいる一方，その膨大な数のために，文書すべてに目を通すには多大な労力を必要とする．

こういった背景から，自然言語で書かれた文書から開発者にとって有益な情報を効率的に抽出するための様々な研究が行われている．対象文書からの情報抽出のために広く用いられるのが，トピックモデルである Latent Dhirichlet Allocation (以下 LDA) [1] [2] である．LDA では文章をトピックに分類することでクラスタリングを行い，文書全体の傾向を探ったり似ている文章を関連付けることで有益な情報を抽出することができる．

Kahai ら [3] は，Eclipse プロジェクトのフォーラムからユーザの投稿を取得し，LDA を用いてトピック分類することで，どのツールのどの機能に関する話題が頻繁に議論されている質問であるかを特定している．また，取得した質問のうち，背景知識の少ない新しいユーザによって質問されているものを手動で抽出している．清ら [4] の研究では，アプリケーション提供サービスのユーザレビューを収集，LDA によってトピック分類し，対象とするアプリケーションの機能ごとに代表となるレビューを抽出し，短期間での低評価レビューの発生件数のデータと組み合わせることで，ユーザがアプリケーションに対して希望している機能の可視化を実現している．Chen ら [5] は，アプリケーションのユーザレビューを言及されている内容によってグループ化し，開発者にとって有益な情報であるかを定式化することでレビューのランク付けを行い，開発者に提示する手法を提案している．我々の先行研究 [6] においては，実ソフトウェアの開発に用いられるテストケース記述を LDA を用いて分類する手法を提案している．

[1] 大阪大学大学院情報科学研究科
Osaka University 1-5 Yamadagaoka, Suita, 565-0871 Japan
[a)] t-hitosi@ist.osaka-u.ac.jp
[b)] nakagawa@ist.osaka-u.ac.jp
[c)] t-tutiya@ist.osaka-u.ac.jp

2. LDA (Latent Dhirichlet Allocation)

本節では，自然言語文書のトピック分類に用いられる LDA が，どのように文書のトピックを決定し，文書分類を可能にしているかを記述する．また，LDA 適用時に発生する問題について説明する．

2.1 トピックモデルとしての LDA

LDA では文章を，トピック混合率に従って各単語の背景トピックが生成され，その背景トピックに従って単語が生成されたものとしてモデル化している．LDA による文章のモデルは図 1 のグラフィカルモデルによって表される．ここで，図中の各変数は以下を表したものである．

α_k: トピック混合率に関するハイパーパラメータ
θ: 文書のトピック混合率

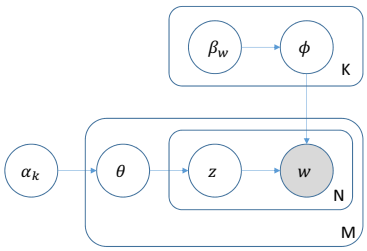

図 1　LDA のグラフィカルモデル

z: 単語の背景トピック
β_w: 単語の生成確率に関するハイパーパラメータ
ϕ: 背景トピック k のときに単語 w が生成される確率
w: 単語 w
K: トピック数
M: 文書数
N: 1 文書中の単語数

文書集合中に文書が M 個，その中の文書 m 中に単語が N 個あり，トピック数が K 個ある場合に，文書 m 中の単語 w を生成することを考える．ユーザが与えるハイパーパラメータ α_k によって，文書 m のトピック混合率 θ が決定され，θ の混合率から単語の背景トピック z が決定される．もう 1 つのハイパーパラメータ β_w より ϕ が決定される．ϕ は背景トピック k の場合に単語 w が生成される確率を表しており，z と ϕ に従って，単語 w が生成される．LDA はこのように文書をモデル化している．

LDA によるトピック分類では各単語の背景トピックを決定した後，それらの混合率から文書のトピックを決定し，

- 各文書のトピック混合率
- 各単語の背景トピック
- 各トピックの主要語

を出力する．各トピックの主要語は，各トピックに分類された単語のうち登場回数の多い単語をまとめたものであり，これを読み取ることで，トピックのおおよその話題を類推することができる．

また，LDA によるトピック分類は教師なし機械学習の一種であるため，学習データを用いないが，予め θ や ϕ，z の定まった文書集合を基に，新しい文書の単語の背景トピックを推論し，新しく追加された文書のトピック混合率を算出することができる．

2.2　ストップワード

LDA での正確なトピックモデリングのために，前処理として対象となる文書集合から分類の妨げとなる単語を除外する．この単語はストップワード [7] と呼ばれ，通常，どの文書でも頻出する単語が選ばれる．ストップワードを除去する操作を組み込んだ LDA によるトピック分類の過程を図 2 に示す．

ストップワードはリストとしてまとめて使われるが，こ

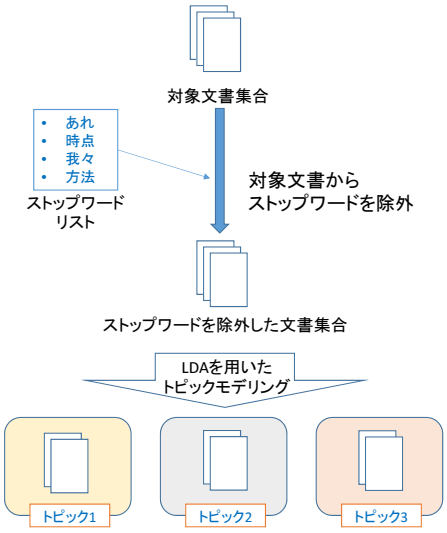

図 2　LDA による文書集合のトピック分類

のストップワードリストは通常，既に完成している物が用いられる．英語や日本語など，言語ごとにストップワードリストが提供されており，Oracle Text[8] では英語やドイツ語など，12 言語のストップワードリストが提供されている．SlothLib[9] では，日本語のストップワードリストが提供されている．SlothLib ストップワードリストには 310 件のストップワードが記載されている．これらのストップワードリストは，その言語での一般的な単語をまとめたものである．しかし，このように一般的な不要語をストップワードとするだけではトピック分類に用いるには不十分である．例えば，テストケース記述のように，“確認”や“押下”など特定の単語を用いた文章が繰り返されるような特徴を持つ文書をトピック分類を行う対象とした場合，対象文書集合中には頻出するが一般的ではない特定の単語が，非常に多くの文書に含まれている場合がある．このようにストップワードに指定すべき語は対象によって異なるため，文書の内容に応じてストップワードを決定し，対象に合ったストップワードリストを用いることが重要である．Moh ら [10] によると，TF-IDF によって単語に重み付けし，文書集合のクラスタリングを行う際に，対象文書からストップワードを決定し，ストップワードを変更することで，重要と思われる単語の重みが大きくなり，クラスタリングがより正確になるとされている．ストップワードを対象によって変更することの有効性を示した一方で，ストップワードとして選ばれる単語は人間の判断により手動で選定しているため，ストップワードを選ぶ操作の自動化を進めていく必要があるといった課題も残っている．

2.3　類似トピック

実際の LDA 適用によるトピック分類においては，単語の構成が類似した複数のトピックが出現する場合がある．これは 1 トピックに分類されるべき文書群が複数トピック

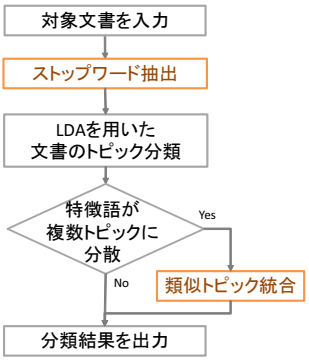

図 3　提案手法の流れ

に分類されてしまうことによって起こるもので，これにより文書分類の精度が下がるという問題が発生する．

3. 提案手法

前節で示した問題を解決するために，本章では，LDA 適用時に前処理と後処理を追加することで，生成されるトピックモデルをより洗練化する手法を提案する．提案手法の全体の流れは図 3 のフローチャートのとおりである．提案手法は LDA の適用前後における以下の 2 つの処理から成り立っている．

Step1: 文書中の単語からのストップワード作成

Step2: 類似トピック統合

以降，これらの処理についてそれぞれ詳細な説明をする．

3.1 対象文書からのストップワード抽出

対象となる文書集合中の全単語について出現率を算出し，出現率が高い単語をストップワードリストに追加することによって，対象文書集合中に特有の頻出語を除去することを可能にする．このようなストップワードの除去を行うことによって，より正確な文書のトピック分類を実現する．各単語の全文書中での出現率の指標として，DF (Document Frequency) を用いる．DF は，文書クラスタリングの特徴量に広く用いられる TF-IDF のうち，一般語のフィルタの役割をもつ IDF (Inversed Document Frequency) を真数に直したものであり，対象となる文書集合中の単語について，式 1 によって算出される．

$$DF_i = \frac{|\{d : d \ni t_i\}|}{|D|} \quad (1)$$

ここで，式 1 中の各変数は，以下を表したものである．

$|D|$: 総文書数

$|\{d : d \ni t_i\}|$: 単語 i を含む文書数

IDF が文書集合中の一般語の特徴量を下げるために最も効果的な指標の 1 つであることから，DF を用いることが文書集合中の一般語の特定に効果があると考えられる．

ストップワード抽出は，疑似コードでは Algorithm1 のように表せる．対象文書集合全体について，各単語の DF を算出し，予め設定した閾値よりも DF が大きい単語を従来のストップワードリストに追加することで，ストップワードリストの改良を図る．

Algorithm 1 対象文書からのストップワード抽出

```
for each words in each topics do
  DF 値を算出
  if DF_w < 閾値: then
    ストップワードリストに追加
  end if
end for
```

3.2 類似トピック統合

類似した文書を多く含む対象を LDA でトピック分類した際，分類結果に似通ったトピックが複数現れ，適切な分類の障害となる場合がある．これを回避するために，類似トピックを統合し，トピックの冗長性を排除する．

LDA を適用した結果から各トピックに分類された単語集合を入力し，各トピックの単語集合に対して，TF-IDF コサイン類似度 [11][12] を利用したクラスタリングを行い，類似したトピックを統合する．コサイン類似度は，文書中の単語の出現回数によって 2 文書間の類似度である．TF-IDF は単語ごとに付加される数値であり，特定の文書にしか出現しない単語の重要度を上げる働きがある．コサイン類似度で用いる各単語について TF-IDF で重み付けをした指標が TF-IDF コサイン類似度である．この操作については，図 3 で示したように，実際に LDA を適用した結果をユーザが見て，必要と判断すれば適用する．

4. 評価実験

本実験の目的は，提案手法を用いたトピック分類と通常のトピック分類で，それぞれ正確性を比較し，手法の有効性を評価することである．

本実験用に，自然言語で記述されたテストケース記述，ユーザレビュー，メーリングリストの 3 種類の文書集合を取得した．各文書集合の特徴を表 4 に示す．テストケース記述は実際の Web アプリケーション開発で使用された文書であり，様々な機能に関するテストケースが日本語で記述されている．トピック分類を行うことで，各機能のテストケース記述が特定のトピックに集まる傾向にあるかを実験によって評価する．ユーザレビューは SNS サービスのスマートフォン用アプリケーションである Facebook for iOS のユーザから投稿されたレビューである．アプリケーション中の様々な機能に言及したレビューが含まれており，トピック分類を行うことで各トピックの単語から言及されている機能を読み取ることが出来るかを評価する．メーリングリストには Apache Commons User List を用いる．Apache Commons User List は Apache Commons に属するソフトウェアのユーザコミュニティとして機能して

表 1　対象文書の概要

対象文書	言語	件数
テストケース記述 (結合テスト)	日本語	142
ユーザレビュー (Facebook for iOS)	英語	3000
メーリングリスト (Apache Commons User List)	英語	1000

表 2　対象テストケース記述

機能	テストケース記述数
共通結合テスト	17
部品受注に関する結合テスト	18
部品引当に関する結合テスト	1
部品出荷に関する結合テスト	6
部品売上に関する結合テスト	5
部品在庫に関する結合テスト	2
部品発注に関する結合テスト	33
部品入荷に関する結合テスト	9
部品バッチに関する結合テスト	9
ラベル受注に関する結合テスト	4
ラベル引当に関する結合テスト	4
ラベル出荷に関する結合テスト	2
ラベル売上に関する結合テスト	2
ラベル在庫に関する結合テスト	2
ラベル発注に関する結合テスト	2
ラベル入荷に関する結合テスト	4
ラベルバッチに関する結合テスト	9
部品請求に関する結合テスト	13

おり，様々なソフトウェアについての質問やアナウンスが書かれた e-mail が含まれている．トピック分類を行うことで各トピックの単語から言及されているソフトウェアや機能を読み取ることが出来るかを評価する．これらの各対象について，次節より実際に行った実験の詳細を述べていく．

提案手法中の LDA によるトピック分類には，LDA を実装したツールとして MAchine Learning for LanguagE Toolkit (MALLET)[13] を用いた．

4.1　実験 1: テストケース記述

4.1.1　対象文書

テストケース記述は，テストにおいて行う操作を自然言語で記述したものである．以下に例を示す．

- A を入力として設定する
- B ウィンドウの表示が C であることを確認する
- C ボタンを押下する
- D を出力する

このようなテストにおける一連の操作をまとめた記述をテストケース記述と呼び，今回評価実験の対象として用いるテストケース記述の機能とテストケース記述数を表 2 に示す．全体のテストケース記述数は 142 件である．

4.1.2　評価基準

提案手法の有効性を評価するために，同機能のテストケース記述が 1 トピックに集約される割合を求め，再現率，適合率，F 値を算出する．再現率は，式 2 のように，機能ごとの 1 トピックに集約された最大テストケース記述数の割合から求められる．また，機能 g のテストケース記述の分類の適合率 $Precision_g$ は式 3 で算出することが出来る．テストケース記述全体での適合率 $Precision$ は式 4 で算出する．これは，各機能での式 3 の計算結果に，機能のテストケース記述数を考慮し重み付けをした上で足し合わせ，全体のテストケース記述数で割った値である．

$$Recall = \frac{\sum_g d_{gt_g}}{|D|} \tag{2}$$

$$Precision_g = \frac{d_{gt_g}}{d_{t_g}} \tag{3}$$

$$Precision = \frac{\sum_g d_{gt_g} \frac{d_{gt_g}}{d_{t_g}}}{|D|} \tag{4}$$

ここで，式 2，3，4 中の各変数は，以下を表したものである．

t_g: 機能 g のテストケース記述が最も多く集約されたトピック

d_g: 機能 g のテストケース記述数

d_{t_g}: トピック t_g に集約されたテストケース記述数

d_{gt_g}: トピック t_g に集約された機能 g のテストケース記述数

$|D|$: 総テストケース記述数

また，再現率と適合率の調和平均を取った値として，F 値がある．

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5}$$

これらの式で表した適合率，再現率，F 値を用いて，ストップワードを用いない場合，通常のストップワードを用いた場合，提案手法を用いた場合での分類結果を比較する．

4.1.3　実験の設定

通常の分類において用いるストップワードリストには，SlothLib ストップワードを使用した．対象文書の単語分割のために MeCab[14] を用いて形態素解析を行った．またこの際，意味の無い単語をフィルタリングするために，抽出する単語の品詞を名詞，自立動詞，自立形容詞，感動詞，副詞，接頭辞，未知語に限定した．単語抽出の結果，テストケース記述は全体で 187,389 語の単語集合となった．

4.1.4　文書からストップワード抽出

文書中の単語の DF を計測した結果，表 4.1.4 のような単語の DF が大きくなった．これらの単語のうち，文書中に頻出するが分類の指標となりにくいと思われる，DF が 0.7 以上の 29 個を通常のストップワードリストに追加することで，新しいストップワードリストを作成した．

作成したストップワードを用いて LDA によるトピック分類を行った．本実験ではトピック数を，表 2 で示した被評価機能数である 18 より十分に多い 30 に設定し，各トピックで登場回数が上位となる主要語を出力した結果，“受注”，“出荷”など，特徴的な語が複数のトピックに散見さ

表 3　DF 上位の単語：テストケース記述

単語	DF	単語	DF
押下	1	更新	0.873239
こと	1	実行	0.866197
確認	1	処理	0.852113
する	1	結果	0.838028
遷移	1	ダンプ	0.838028
表示	1	入力	0.830986
画面	1	条件	0.816901
リンク	0.985915	登録	0.802817
ボタン	0.978873	前後	0.78169
情報	0.957746	明細	0.739437
照会	0.950704	後	0.725352
検索	0.922535	指示	0.711268
データ	0.915493	出力	0.711268
メニュー	0.915493	在庫	0.704225
選択	0.894366	修正	0.697183

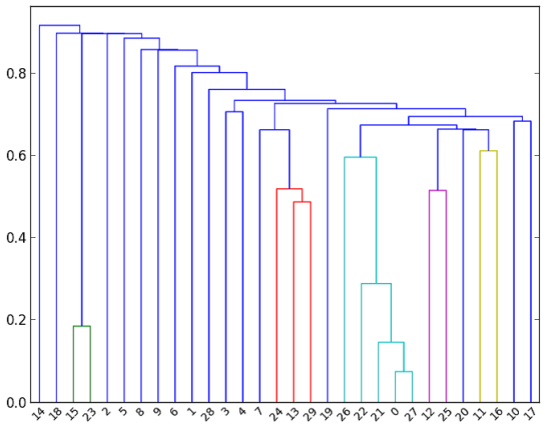

図 4　トピック間距離のデンドログラム：テストケース記述

れ，主要語の類似したトピックが出現した．そこで，実験の次の段階として，図 3 で示した提案手法のフローに従い，類似トピックの統合を行う．

4.1.5　類似トピック統合

トピック 0 から 29 までの全ての単語を入力とし，トピック間の距離を算出する．トピック間の距離をデンドログラムとして出力した結果を図 4 に示す．横軸はトピック ID，縦軸はトピック間距離を表しており，単語の分布が類似したトピックでは距離が近くなっていることが分かる．最も距離の近いトピック 0 とトピック 27 には，どちらも受注に関するテストケース記述が集まっており，“受注”，“保存”など共通の単語が含まれている．また，それらと距離の近いトピック 21，22 などでも“受注”が主要語集合の上位に入っており，受注に関するトピックがまとめられている．このデンドログラムを参考に，トピック間の距離が 0.6 未満であるトピック群を類似トピックと判断し，統合を行う．今回の実験では，トピック 0+21+22+26+27，トピック 15+23，トピック 13+24+29，トピック 12+25，トピック 11+16 が統合され，その上で各テストケース記述のトピック混合率の再計算を行った．

表 4　テストケース記述分類の再現率，適合率，F 値

	ストップワード無し	通常	提案手法
再現率	0.634	0.641	0.718
適合率	0.454	0.469	0.548
F 値	0.529	0.542	0.622

4.1.6　結果

実験結果として再現率，適合率，F 値を表 4.1.6 に示す．提案手法を用いた場合，ストップワード無し，通常よりも再現率，適合率ともに高い値を得られ，F 値も改善される結果となった．

4.2　実験 2: ユーザレビュー

4.2.1　実験対象

本実験の対象として，実際のアプリケーション配信サービスに投稿されたユーザレビューを用いた．2015 年 8 月 5 日から 9 月 8 日までに米国の App Store に投稿された，Facebook のユーザレビュー 3,000 件を取得し，タイトルと本文を抽出する．以下に取得したユーザレビューの一例を示す．

> Title: Buggy, navigation is not good, too many ads
> Body: ... And the ads that are ”suggested” posts have no relevance to me.

このユーザレビューでは，本文で Facebook アプリ中の要素である広告について主に言及し，広告のサジェスト機能が自分に合わない内容であると述べている．

4.2.2　評価基準

本実験では，通常のストップワードを用いた場合と提案手法を用いた場合の分類結果を比較し，提案手法を用いることで，トピックの主要語集合から各トピックで言及されている特定の機能が読み取りやすくなっているかを評価する．

4.2.3　実験の設定

対象となるユーザレビューは英語で記述されているため，通常の分類で用いるストップワードリストには Oracle Text の英語ストップワードリストを用いた．3000 件のユーザレビューを単語分割し，全体で 67,882 語の単語集合となった．

4.2.4　文書からストップワード抽出

文書中の単語の DF を計測した結果，上位には“the” (DF = 0.589)，“i” (DF = 0.543) など一般的な単語が含まれていた．これらの単語は通常の英語ストップワードリストに含まれているが，“app” (DF = 0.46)，“facebook” (DF = 0.268) など，Facebook ユーザレビューに頻出する単語も比較的 DF が高くなった．文書中に頻出するが分類の指標になりにくいと考えられる DF が 0.1 以上の 42 語を通常のストップワードリストに追加することで，新しいストップワードリストを作成した．

表 5　トピックの機能ラベル：ユーザレビュー

通常		提案手法	
トピック	機能	トピック	機能
0	写真	0	通知
1	要望	1	フリーズ
2	不明	2	アップデート
3	ボタン	3	アップデート
4	アップデート	4	メッセンジャー
5	メッセンジャー	5	アカウント
6	アップデート，フリーズ	6	写真，クラッシュ
7	スペイン語レビュー	7	不明
8	不明	8	要望
9	不明	9	ビデオ
10	ビデオ	10	ニュースフィード
11	画面	11	フレンド
12	ニュースフィード	12	バッテリ
13	フリーズ，クラッシュ	13	アップデート
14	ページ	14	スペイン語レビュー
15	バッテリ	15	クラッシュ，フリーズ
16	コメント	16	ページ
17	ビデオ，写真	17	広告
18	アカウント	18	評価するレビュー
19	評価するレビュー	19	写真

4.2.5　結果

作成したストップワードを用いて LDA によるトピック分類を行った．本実験ではトピック数を 20 に設定し，通常の分類と提案手法を用いた分類について，各トピックで登場回数が上位となる主要語を出力し，それぞれについて手動でトピックのラベル付けを行った．結果を表 4.2.5 に示す．通常の分類では，主要語から言及されている機能が特定できないトピックがトピック 2，トピック 8，トピック 9 の 3 個であったのに対し，提案手法ではトピック 7 の 1 個だけとなり，改善されている．また，主要語集合では単語が出現回数順に整列しているため，先頭にある単語ほど重要であると考えられるが，通常の分類では“app”，“facebook”など一般的な単語が出現していた．これに対し，提案手法を用いた分類では，より特徴的な単語が先頭に現れるようになるという結果が得られ，トピックの主要語の可読性が向上した．また，4.2.1 節において具体例として示したユーザレビューは，通常の分類ではボタンに関するトピック 3 に分類されていたが，提案手法を用いると広告に関するトピック 17 が形成され，分類結果が改善された．このようにトピックごとの機能が読み取りやすくなったことで，実際に開発者がユーザレビューを参照する場合に，言及されている機能の判別が短時間で行えるようになると考えられる．

4.3　実験 3: メーリングリスト

4.3.1　実験対象

本実験の対象として，Apache Commons User List[15] に投稿された e-mail を用いた．Apache Commons User List は，Apache ソフトウェア財団が管理している java コンポーネントユーザ間の連絡のために使われるメーリングリストである．投稿される e-mail の大多数は Apache Commons に属するソフトウェアを使用した際の技術的な質問とそれに対する返答であり，ユーザフォーラムとして機能している．また，その他にソフトウェアの最新バージョンリリースのアナウンス，バグ報告，ソフトウェア仕様についての質問などの e-mail が含まれている．2002 年 9 月から 2003 年 1 月までに投稿された，メーリングリスト中の e-mail を 1,000 件取得し，差出人タイトルと本文を抽出する．以下に取得した e-mail の一例を示す．

> Subject: [Jelly] Import and Include tags
> What are the differences between the import and include core tags? When to use one vs the other?

このメールには，Apache Commons に属するスクリプトエンジンである Jelly について，“Import”と“Include”のタグの使い分けについての質問が書かれている．このように，ユーザコミュニティとして用いられるメーリングリストは，多くのユーザが使いづらいと思っている機能や，特定のバージョンでのバグ報告が含まれており，これらを解析することで開発者にとって有益な情報が得られる文書集合である．

4.3.2　評価基準

本実験では，対象となるメーリングリストについて，LDA によるトピック分類を行い，メール中で言及されているソフトウェアごとにトピック分類する．通常の分類結果と提案手法を用いた分類結果を比較し，提案手法を用いることで，トピック分類がより正確になっているかを，トピックの主要語集合に登場する単語を用いて評価する．

4.3.3　実験の設定

実験 2 と同様に，通常の分類で用いるストップワードリストには Oracle Text の英語ストップワードリストを用いた．1,000 件の e-mail を単語分割し，全体で 67,882 語の単語集合となった．

4.3.4　文書からストップワード抽出

文書中の単語の DF を計測した結果，“is”，“a”などの一般的なストップワードに加えて，“re”，”date”などの件名や投稿日時やなどの表記に使われている単語の DF の値が高くなっていた．これらの単語のうち，文書中に頻出するが分類の指標になりにくいと考えられる DF が 0.1 以上の単語 233 件を通常のストップワードリストに追加することで，新しいストップワードリストを作成した．作成したストップワードを用いて LDA によるトピック分類を行った．獲得トピック数はメール中で言及されると考えられるソフトウェア数より十分に多い 20 とした．

4.3.5　類似トピック統合

トピック 0 から 19 までの全ての単語を入力とし，トピック間の距離を算出する．トピック間の距離をデンドログラムとして出力した結果を図 5 に示す．

また，統合前のトピック主要語集合を表 4.3.5 に示す．最も距離の近いトピック 1 とトピック 13 の主要語を表 4.3.5

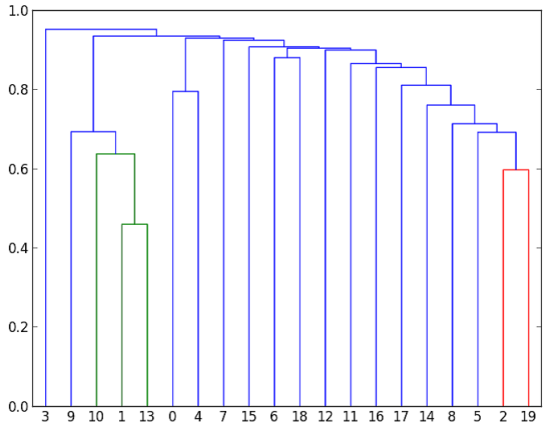

図 5　トピック間距離のデンドログラム：メーリングリスト

で見ると，先頭付近に“jelly”が含まれており，これらは両方とも Apache Jelly に関するメールが分類されたトピックであると考えられる．また，デンドログラムを見るとトピック 9，10，1，13 が他トピックとは離れてクラスタリングされている．これらのトピック全ての主要語には“jelly”が登場しており，他トピックの主要語集合には含まれていなかった．よって，Apache Jelly に関するトピックが 1 つにまとまったと考えられる．また，トピック 2 とトピック 19 は JDBC Utility Component に関するトピックであるが，トピック間距離が小さく，類似トピックで統合され，JDBC に関するトピックが 1 つにまとまった．

4.3.1 節において具体例として示した e-mail は，通常の分類では Apache Validator に関するトピックに分類されていたが，提案手法を用いた場合では Apache Jelly に関するトピックに分類されており，分類結果が改善された．

5. 考察

5.1 評価結果

実験 1 では提案手法が通常の分類よりも分類精度が高かった結果について，提案手法中のストップワードリストに単語を追加する行程が特に貢献したと思われる．本実験で用いたテストケース記述のように，一般的ではないが特定の単語が全文書中に頻出するという特徴を持った自然言語文書群を分類対象とする場合，これらの単語がノイズとなってしまう．これが，通常のストップワードを用いた場合での分類結果がストップワードリストを用いない場合の結果よりも改善されなかった実験結果の要因である．提案手法によりストップワードを追加し適当な不要語が除外されたことで，LDA がより特徴的な語を認識しやすくなり，分類結果が改善されたと考えられる．実験 2，3 では，通常の分類では誤ったトピックに分類されていた文書が提案手法によって正しいトピックに分類された．これはストップワード抽出により不要語が取り除かれたことで分類の精度が向上したためである．実験 1，2，3 の結果より，様々なドメインの文書分類に対して提案手法が有効であるといえる．

表 6　トピックの主要語集合：メーリングリスト，提案手法

トピック	主要語
0	public void idx protected class bigdecimal return boolean bean timestamp values long java object attributes org double string apache
1	**jelly** paul tag org apache tags var set foo script output libbrecht run xmlns java impl class activemath xml
2	org java apache jar dbcp user yahoo cli mysql mail **jdbc** http class mailto garrett gmane smith command john
3	jxpath test string getvalue dmitri xforms pointer context label configuratorcontext ldap user jxpathcontext daniel xml return org book field
4	betwixt apache org junit java class bean test xml abstractbeanwriter user beanreader xmlintrospector thread debug lang vfs martin torque
5	digester object string properties public property bean xml rule user file connecturi org class element help apache mailto key
6	java org apache catalina core source unknown invoke error net postgresql logging yahoo lang connection trace io reflect logfactory
7	apache org map user mailto object mail collections integer java stephen int listutils dynabean maputils help tomap predicatedlist moritz
8	mailto user mail apache org help digester october original users pm scott mark comp parameter nbsp rule gmane friday
9	test **jelly** knut null expected user wannheden mail xmlunit isempty tag org stringutils parse isspaces xu lang gmane comp
10	james yahoo strachan http uk **jelly** xml file need ll page user weblogs web radio music sport news charts
11	user comp gmane will could craig ve time work don make need ll different release class bug implementation good
12	org apache user validator mailto file help mail form http ftp gmane comp fileupload field problem tan license release
13	error javac **jelly** org apache ant java class maven symbol sandbox tags src peter home apps location resolve cannot
14	digester java org apache jar xerces xml servlet parse lang xmldocumentfragmentscannerimpl region impl validator web parsers nosuchmethoderror parser error
15	beanutils string java public org apache bean propertyutils custom lang class util void object certification user seacor problem return
16	httpclient user option org apache id http string post options header column create system main robert application problem help
17	digester object person mail intended john please builder information stack confidential httpclient scioworks build organisation sender pop recipient will
18	log logging apache org struts thread tomcat logger user debug class file jar messages safety properties appender web classes
19	pool connection dbcp apache org **jdbc** user parameter tomcat driver datasource connections java null getconnection sql conn comp idle

5.2 提案手法の適用範囲

本手法中の，文書からのストップワード抽出，類似トピック統合の両処理は，同じドメインの文書を集めた対象文書集合に有効である．これは，同じドメインの文書群にはある程度の頻出語が登場するという前提があるためであり，同じ単語が頻出しない文書集合に適用する場合には，分類結果の改善は期待できない．例えば，全く別のドメインから少数ずつ文書を集めた文書集合をトピック分類し，関連性を発見するといった問題設定などでは，本手法は通常の分類と同程度の正確性になり効果を発揮することができないと考えられる．

2.1 節で述べたように，LDA では既存の分類結果を教師データとして新規文書のトピックを推定することができる．実際のトピック推定では，新規文書には教師データの文書集合と同じドメインに属する文書を用いると考えられるので，本手法で作成したストップワードリストを用いることが有効である．

ストップワード抽出の際には DF 上位の単語を除外する

が，この閾値は 3 種類の実験から対象によって適切な値が異なることが分かった．テストケースのように各文書中に似ている文を多く含む対象文書では，DF があまり低くないが文書の特徴を表すと考えられる語も出現するため，閾値は高めに設定する必要がある．また，ユーザレビューやメーリングリストなど，より口語に近い対象文書では DF の高い語は比較的少なく，閾値は低めに設定した方が良い．トピック統合の閾値についても同様で，ユーザがトピック間距離のデンドログラムと主要語集合の構成単語を参照し，統合の閾値を手動で設定する方法が現時点では最も妥当な統合結果を得られると思われる．これらの閾値設定はユーザの感覚に基づいた判断に頼るため，ユーザのトピック分類に関する背景知識を必要とするほか誤った閾値設定をすると重要な語がストップワードとなり，通常の分類よりも精度が落ちてしまう．こういった問題から，閾値設定の自動化が今後の課題である．

5.3 さらなる分類精度向上のために

本実験では，4.1.3 小節で示したように，日本語の対象文書に対して形態素解析を行い，品詞によって不要語を予め取り除いている．これに対し，英語の対象文書は単語分割を行うのみに止まっている．これにより，英語で書かれた Facebook for iOS レビュー，Apache Commons User List をそれぞれ対象とした実験 2,3 では，通常のストップワードリストに既に含まれている語が DF 上位の単語に多数出現していた．英語文書に対しても形態素解析を行い，日本語の対象と同じようにトピック分類の対象とする品詞を限定する操作や，活用語のステミングを行うなど，前処理の段階で不要語を取り除いておくことで，分類対象の単語集合がより洗練化され，LDA によるトピック分類をさらに正確にすることができると考えられる．

6. まとめ

本稿では，LDA を自然言語文書に適用する際に発生する問題を特定し，対象となる文書集合からストップワードを抽出し通常のストップワードリストに追加することで，対象文書に適切なストップワードリストを作成する手法を提案した．また，分類したトピックの類似度を算出することで，トピック分類の正確性を向上させる手法も同時に提案した．これらの手法を組み合わせることで，通常の分類手法と比較して，分類の正確性が向上していることが 3 種類の文書集合を対象とした評価実験によって確認できた．今後の課題としては，ストップワード抽出とトピック統合の閾値設定の自動化や，対象文書の構文解析や形態素解析によるステミングの強化などによって言語による分類性能の差異を吸収する手法の考案などがある．

謝辞 本研究に際して，テストケースデータと分析に関する知見を提供くださいました，サントリーシステムテクノロジー株式会社長谷川壽延様，森嶋崇様に深く感謝いたします．

参考文献

[1] Blei, D. M., Ng, A. Y. and Jordan, M. I.: Latent Dirichlet Allocation, *J. Mach. Learn. Res.*, Vol. 3, pp. 993–1022 (2003).
[2] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P. and Welling, M.: Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'08)*, pp. 569–577, New York, NY, USA (2008), ACM.
[3] Kahani, N., Bagherzadeh, M., Dingel, J. and Cordy, J. R.: The Problems with Eclipse Modeling Tools: A Topic Analysis of Eclipse Forums, in *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems(MODELS'16)*, pp. 227–237, New York, NY, USA (2016), ACM.
[4] 清雄一, 田原康之, 大須賀昭彦：レビューサイトの情報を利用したスマートフォンアプリケーションの開発支援, 情報処理学会研究報告. ソフトウェア工学研究会報告, Vol. 2014, No. 4, pp. 1–8 (2014).
[5] Chen, N., Lin, J., Hoi, S. C. H., Xiao, X. and Zhang, B.: AR-miner: Mining Informative Reviews for Developers from Mobile App Marketplace, in *Proceedings of the 36th International Conference on Software Engineering(ICSE'14)*, pp. 767–778, New York, NY, USA (2014), ACM.
[6] 高橋仁, 中川博之, 土屋達弘：文書からのストップワード抽出によるトピックモデル洗練化 (知能ソフトウェア工学), 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報, Vol. 116, No. 284, pp. 19–24 (2016).
[7] Rajaraman, A. and Ullman, J. D.: *Mining of Massive Datasets*, Cambridge University Press, New York, NY, USA (2011).
[8] Oracle Text で提供されるストップリスト, `https://docs.oracle.com/cd/E16338_01/text.112/b61357/astopsup.htm`.
[9] 大島裕明, 中村聡史, 田中克己：SlothLib: Web サーチ研究のためのプログラミングライブラリ, 日本データベース学会 Letters, Vol. 6, No. 1, pp. 113–116 (2007).
[10] Moh, T.-S. and Bhagvat, S.: Clustering of Technology Tweets and the Impact of Stop Words on Clusters, in *Proceedings of the 50th Annual Southeast Regional Conference(ACM-SE'12)*, pp. 226–231, New York, NY, USA (2012), ACM.
[11] Tata, S. and Patel, J. M.: Estimating the Selectivity of Tf-idf Based Cosine Similarity Predicates, *SIGMOD Rec.*, Vol. 36, No. 2, pp. 7–12 (2007).
[12] Whissell, J. S. and Clarke, C. L.: Effective Measures for Inter-document Similarity, in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management(CIKM'13)*, pp. 1361–1370, New York, NY, USA (2013), ACM.
[13] MALLET homepage, `http://mallet.cs.umass.edu/`.
[14] MeCab : Yet Another Part-of-Speech and Morphological Analyzer, `https://www.mlab.im.dendai.ac.jp/~yamada/ir/MorphologicalAnalyzer/MeCab.html`.
[15] Apache Commons Mailing Lists, http://commons.apache.org/mail-lists.html.