CF-012

圧縮距離と分散表現を用いた電子メールのクラスタリング

佐藤 哲 †

NHN テコラス株式会社 †

1. はじめに

電子メールのコンテンツ分析には，長い歴史と多くの研究がある．しかし，異常検出の一種と言える spam メール検出技術は既に効果も普及率も高いことが知られているが，電子メールコンテンツの多様性から，より一般的に自動分類することは難しい．そこで本研究では，データの多様性に強い圧縮距離に基づくクラスタリング手法と，データをもとに学習する分散表現による類似度測定手法を組み合わせることで，コンテンツの多様性に対応してクラスタリングを実現する手法を提案する．提案手法は，学習データを必要とする・しない手法と，入力データの量を必要とする・しない手法が組み合わされており，各手法の短所を互いに補完することを目指している．

2. 電子メールコンテンツに対するクラスタリング手法

電子メール，とりわけ技術系企業が受信するメールは単なるテキストメッセージに留まらない多様性があり，コンテンツ分析が困難である．まず，電子メールの規格に由来する要因として，複数の文字コードや言語，base64 などのエンコード，HTML メールによる記号の混入，添付ファイルの存在などがある．前処理によりそれらの問題を軽減させたとしても，メールが様々なデータの送信に使われることに由来する要因として，人間同士の自然言語によるコンテンツ，システムログやシステム警告のような機械生成されたコンテンツ，スパムやフィッシングと呼ばれる不必要なコンテンツなどの内容の多様性がある．さらに，メッセージの引用や転送によるコンテンツのさらなる混在もあり得る．これらの問題のため，多くの関連研究ではメールがある程度の構造を持つこ場合を仮定してる [1][2].

このように非構造でノイズの多いデータに対する効果的な距離尺度として知られているものの一つに圧縮距離 (NCD; Normalized Compression Distance) がある [3]．圧縮距離はその汎用性から様々なデータ系列の分類に利用されており，電子メールのコンテンツに対しても適用可能だと考えられる．しかし圧縮距離はデータの圧縮率を利用しているため，圧縮効果が期待できない短い文章のようなデータには適用できない．短い文章に適用可能な距離尺度として利用できる手法としては，単語や文章にベクトルデータを対応させる word2vec[4] のような分散表現技術があるが，分散表現を得るためには学習データが必要である．人間が生成した大量の文書から学習して分散表現を得る手法もあるが，予めラベルが付いた学習データから分散表現を得る手法の方が精度が高い場合がある．ただし学習データを人間が作成することはコストが高いため，自動的に作成可能であることが求められる．そこで本研究では，圧縮距離を規準としたコンテンツ比較により一部のデータに対しクラスタリングを実施し，クラスタ ID とクラスタコンテンツデータのペアを学習データとする．学習データ対し Paragraph Vector[5] を適用することによって分散表現を作成し，入力データの分散表現と学習データであるクラスタリング済みデータの各クラスタとの類似度を計算することで，分類問題として入力データを逐次的にクラスタリングするアルゴリズムを提案する．

Clustering Email Contents using Compression Distance and Distributed Representation

†Tetsu R. Satoh，NHN Techorus Corp.

3. 圧縮距離と分散表現

圧縮距離と分散表現は，共に非構造化データ同士の距離を測定可能にする技術と言える．

まず圧縮距離は，データの複雑性をランダムネスによって定義し，それを比較することによってデータ同士の距離を定義する手法である．圧縮距離は任意のデータ系列に対し，以下のように定義される：

$$NCD(x,y) = \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

ここで，$Z(x)$ はデータ x を圧縮した後の長さであり，$Z(xy)$ はデータ x と y を結合して圧縮した後の長さである．圧縮処理が使われるため，計算コストは高い．しかしノイズに強くデータの種類によらない汎用性があり，学習アルゴリズムではないので学習データは必要ない．

一方で分散表現は，データ群から確率分布を推定し，個々のデータをベクトル表現する手法である．多くの実現方法が考えられるが，参考文献 [5] では次のような平均対数確率密度関数を最大化するパラメータを推定する問題に定式化されている：

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(\omega_t | \omega_{t-k}, \cdots, \omega_{t+k})$$

ここで，T は単語数，k はある単語に関連する文脈近傍の単語数を表す定数，ω_i は単語である．つまり，

ある単語の周辺の単語からその単語が出現する確率を求めるモデルである．文章中で単語の近傍に存在する単語を重要視することはいわゆる分布仮設の考え方が元になっており，学習データが仮定に基づかない場合は適用が難しい．例えばランダムに単語を埋め込まれた spam メール文書や，意味的には関連性のない単語が埋め込まれたシステムログメール文書などに対しては学習が困難である．また，確率分布の推定には通常ニューラルネットワークが用いられるため，入力データの量が求められ計算コストは高い．しかし学習モデルが作成できれば，圧縮距離のように入力データのサイズの大きさは必要とされない．数バイトの単語同士の類似度も計算可能である．

以上のように，圧縮距離と分散表現は性質が大きく異なり，組み合わせることでそれぞれの短所を補完しあえる可能性がある．

4. 提案手法

提案手法では，圧縮距離が適用できないデータ間の距離計算には分散表現を用い，分散表現自身では生成が難しい学習データ作成には圧縮距離を用いる．

提案するクラスタリング手法の手順は以下である：

1) 電子メールデータに対し，文字コードの統一，デコードなどの前処理をする
2) 前処理済みのデータより一部を抽出し，以下のようにクラスタを作成する
   a) 学習データとする電子メールデータ間の圧縮距離を計算し，類似度行列を作成する
   b) 類似度行列に基づきクラスタリングし，クラスタリング済みデータとする
3) クラスタ ID とクラスタコンテンツを入力とし，学習により各クラスタの分散表現を得る
4) 入力データとしての電子メールデータとクラスタリング済みデータの分散表現同士の類似度を計算し，電子メールを分類する

ここで，2) にてデータの一部を抽出しているのは，全データに対し圧縮距離を計算することはコストが高いこと及びサイズの小さなデータに対しては圧縮距離は精度が落ちることの 2 つの理由による．類似度行列 W の成分 w_{ij} は，電子メール i と j の本文文字列 s_i と s_j より

$$w_{ij} = \max(1.0 - NCD(s_i, s_j), 0.0) \qquad (1)$$

により計算する．max 関数が必要なのは，理論的には $[0, 1]$ の値を取る圧縮距離が，近似誤差により 1.0 を超えることがあるためである．クラスタリングは，任意の類似度行列に対し適用可能なグラフクラスタリングを採用する．3) での学習では 2) により用意されたクラスタリング済みデータのクラスタ ID とクラスタコンテンツのペアを学習データとし，学習モデルを生成する．また 4) では，逐次的に入力される電子メールデータをクラスタリング済みデータに対する分類問題に帰着させることでクラスタリングを実施する．分類は，入力データとクラスタリング済みデータの分散表現同士の類似度が最も高いものを対応させることにより実現する．これらの処理の概要を図 1 に示す．

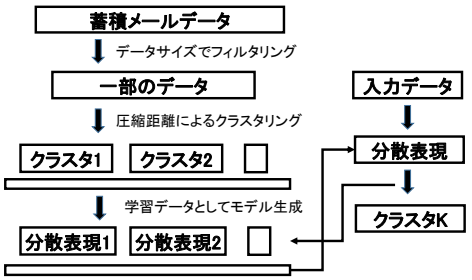

図 1: 処理の流れ

5. 実験

ここでは，実際の電子メールを入力として実験する．電子メールデータは 2019 年 1 月の電子メールデータで，ファイル数は約 5 万通，合計のファイルサイズは約 1G バイトである．ただし以下では，予備実験として，その中からランダムにサンプリングした学習用とテスト用それぞれ 100 通ずつのメールを対象とした実験結果を述べる．

実験では，まず企業のあるメールアドレスに送信されてきた電子メールに対し，POP サーバからメールを取得して base64 のデコードや文字コードの UTF8 化を行ったあと，JSON Lines 形式で Apache Hadoop[†] の HDFS にメールデータを蓄積するシステムを構築した．その後，Apache Spark[††] を用いてデータ読み込みからクラスタリングまでの一連の処理を実行した．

まず，学習データ作成時には，既に述べたように圧縮距離はサイズが大きなデータにのみ適用可能なので，1 通のファイルサイズが 4k バイト以上であることを条件にメールを抽出した．そして式 (1) に基づき 100 通同士の類似度を計算し，100×100 の大きさの類似度行列を作成する．圧縮距離の実装には Apache Commons Compress[†††] の xz アルゴリズムを用いた．そして類似度をグラフの辺の重みとして，類似度行列からグラフクラスタリングを実行する．グラフクラスタリングの実装は，マルコフクラスタリング [6] の実装である MCL Spark[††††] を用いた．マルコフクラスタリングの主なパラメータを表 1 に示す．実験では，クラスタの数を示す規準はマルコフクラスタリングのパラメータである expansion/inflation により定まるものを用いた．その結果，学習用電子メールをクラスタリングして表 2 のような結果が得られた．

[†] https://hadoop.apache.org/
[††] https://spark.apache.org/
[†††] https://commons.apache.org/proper/commons-compress/
[††††] https://github.com/joandre/MCL_spark/

表 1: マルコフクラスタリングの主なパラメータ

expansion	2
inflation	1
iteration	10

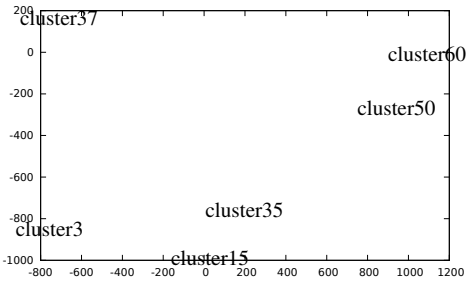

図 2: 圧縮距離によるクラスタリング結果の可視化例

より詳細なクラスタリングが可能であるが，ここでは説明のため表の 6 クラスタのみを用いる．クラスタの ID は単なる識別子であるので，数字には意味がない．　ここで，表 2 のクラスタ要素内容については，ログ監視メールはサーバログの更新データ送信メールであり，メール送受信エラーが中心であった．spam メールは言語により大きく 3 つに分かれた．顧客対応メールは人間同士のメールのやり取りであるので日本語文書である．アラートメールはシステム障害の通知であり，ディスク障害検出通知が中心であった．このクラスタリング結果のデータに対し分散表現を求め，t-SNE を用いて可視化すると図 2 のようになる．t-SNE の実装には Deep Learning for Java† を用いた．図 2 では，多量の日本語文章である日本語 spam メール（クラスタ 37）と，自然言語が含まれていないアルファベットや記号の内容であるクラスタ 3/15/35 が離れるなどコンテンツの分離が確認できる．

次に，得られたクラスタリング結果において，クラスタ ID とクラスタ要素であるメールコンテンツに対し Paragraph Vector を適用して分散表現モデルを得る．Paragraph Vector の実装にも Deep Learning for Java を用いた．Paragraph Vector の学習のための主なパラメータを表 3 に示す．

最後に，この分散表現モデルに基づき電子メール 100 通をテストデータとして，各クラスタとの類似

表 2: 圧縮距離からのクラスタリング例

ID	クラスタ要素内容
3	ログ監視メール
15	中国語等による spam メール
35	英語による spam メール
37	日本語による spam メール
50	顧客対応メール
60	アラートメール

†https://deeplearning4j.org/

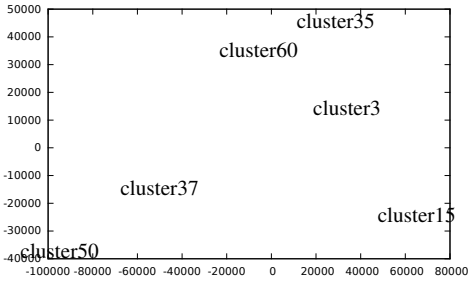

図 3: 分散表現の類似度によるクラスタリング結果の可視化例

度を測定し，最も類似するクラスタの ID を付与する逐次型のクラスタリングを行った．作成されたクラスタに対し分散表現を求め，t-SNE を用いて可視化すると図 3 のようになった．この結果からは，図 2 とは異なり日本語文書であるクラスタ 37 と 50 や，アルファベットのみのコンテンツであるクラスタ 35 と 60 が近づくなど，自然言語としての特徴が現れている．コンテンツを考えると，spam メールであるクラスタ 37 と顧客対応メールであるクラスタ 50 は文章が全く違うため，離れることが望ましい．問題としては，コンテンツを手動で確認し分類の正解率を求めると約 50%であったことがあげられる．また，基準となるクラスタデータについてもコンテンツ分類の精度は完全でなかったり，コンテンツに複数のクラスタの内容が含まれる場合などクラスタリング一般の問題もあり，多くの課題が残っていることが分かった．

6. 考察

実験の結果より，提案手法を用いてもクラスタリング精度には改良の余地があることが分かった．ここでは，クラスタリング精度に影響の大きい類似度計算について考察する．

提案手法では分散表現の学習に用いるクラスタリング済みデータを圧縮距離による類似度計算によって求めているが，電子メールデータから分散表現を求め，分散表現の類似度計算に通常用いられるコサイン類似度を用いてクラスタリングすることも可能である．しかし，電子メールデータに対しては，クラスタリングのパラメータによってクラスタ数が 1 であったり多量であったりと安定しなかった．確認するために，電子メールコンテンツ同士の類似度を圧縮距離を用いて計算した結果の可視化例を図 4 に，分散表現のコサイン類似度により計算した結果の可

表 3: Paragraph Vector の主なパラメータ

iteration	5
ephoch	1
layer size	100
learning rate	0.025
window size	5

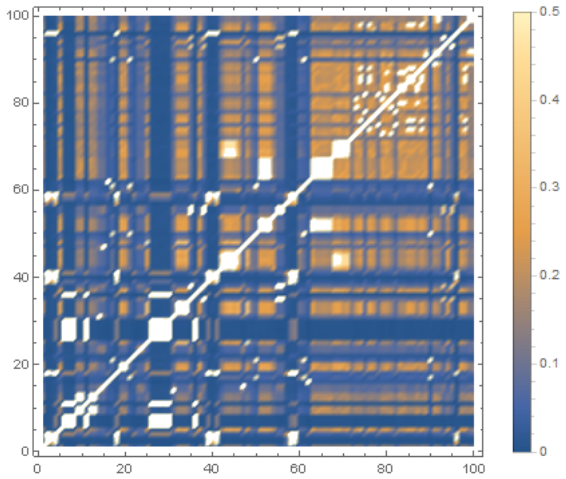

図 4: 圧縮距離による類似度行列の可視化例

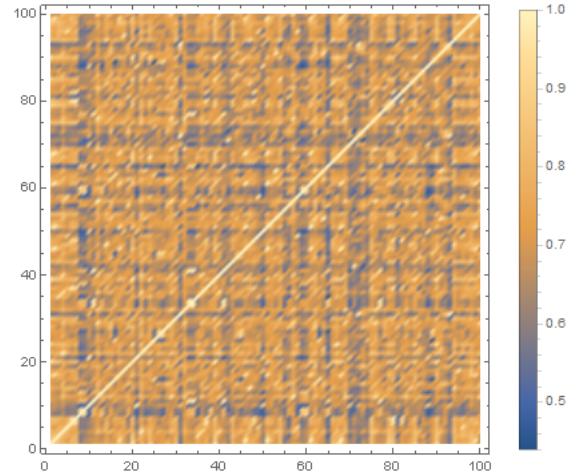

図 5: 分散表現のコサイン類似度による類似度行列の可視化例

視化例を図 5 に示す．なお，マルコフクラスタリングにおいては類似度はグラフの辺の重みであり非負でなければならないので，コサイン類似度はシフト化（$similarity \rightarrow (similarity + 1.0)/2.0$）している．図は類似度行列を濃淡により可視化したものであり，濃淡差があるほどクラスタリング処理をしやすいことを表す．図より，分散距離のコサイン類似度を用いた場合は圧縮距離を用いた場合に比べ濃淡差が少なく，処理が難しいことが分かる．

また，クラスタリング済み電子メールデータとの比較による入力データ分類時の失敗を確認するため，比較の際に参照したコサイン類似度の値のヒストグラムを図 6 に示す．図より，多くの場合は類似度が 0.5 の近辺にあり明確な差が無いため，識別が難しい傾向があることが分かる．例えば，クラスタ 50 の顧客対応メールであるはずのメールデータは，高い確率でクラスタ 37 の日本語 spam メールに分類されていたが，その場合の類似度の差は非常に小さかった．

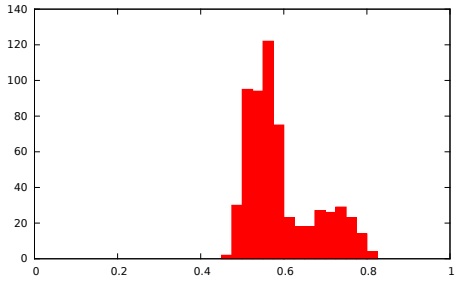

図 6: 分散表現によるクラスタとの類似度結果ヒストグラム

この原因は，コサイン類似度自体の特性，分散表現によるベクトルデータの次元の大きさの問題，日本語の分かち書きの困難性など多くのものが考えられ，今後の研究課題である．

7. おわりに

本研究では，電子メールをコンテンツに基づきクラスタリングするために，圧縮距離を用いて一部のデータをクラスタリングし，クラスタリング済みデータと入力データを分散表現の類似度で比較し分類することで，電子メールデータを逐次的にクラスタリングする手法を提案した．実験により，圧縮距離では計算できないデータに対しても分散表現を利用することで，精度に課題はあるもののクラスタリング結果を得ることができた．また，提案手法の問題点についても議論した．

参考文献

[1] Y. Sun, L. Garcia-Pueyo, J. B. Wendt, M. Najork and A. Broder, Learning Effective Embeddings for Machine Generated Emails with Applications to Email Category Prediction, Proc. Int'l. Conf. Big Data, Vol. 1, pp. 1846–1855, 2018.

[2] F. Kocayusufoglu, Y. Sheng, N. Vo, J. Wendt, Q. Zhao, S. Tata and M. Najork, RiSER: Learning Better Representations for Richly Structured Emails, Proc. WWW Conf., pp. 886–895, 2019.

[3] R. Cilibrasi and P.M.B. Vitanyi, Clustering by compression, IEEE Trans. Inform. Theory., Vol. 51, pp. 1523–1545, 2005.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed Representations of Words and Phrases and Their Compositionality, Proc. 26th Int'l Conf. Neural Information Processing Systems, Vol. 2, pp. 3111–3119, 2013.

[5] Q. Le and T. Mikolov, Distributed representations of sentences and documents, Proc. 31st Int'l. Conf. Machine Learning, pp. 1188–1196, 2014.

[6] S. van Dongen, Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht, 2000.