音声によるツイート再生アプリケーションの開発
Development of tweet playback application

重松 和明
Kazuaki Shigematsu
法政大学情報科学部ディジタルメディア学科
kazuaki.shigematsu.7j@stu.hosei.ac.jp

abstract

Auditory methods are used to understand the contents of text. There are advantages such as hands-free and eyes-free if the contents can be grasped by hearing. Using hearing to understand text take time unlike vision which can look over text. In this research, we focus on Twitter. User is presented with a list of topics generated from the timeline, and select some topics. Then the application plays tweets related to the topics. This reduces the difference between the time spent on vision and hearing. In this research, we seek a method that enables topic classification in each individual's timeline. We conducted experiments using conventional methods for news and Twitter. I used k-means, 77% of tweets about a topic could be collected into one cluster. However, it has not been possible to propose a method to specify what the cluster is like. As a solution, there is a method to indicate with multiple words.

1 まえがき

携帯端末が発展する事で外出先での情報の閲覧が容易になった. 一方, Twitter[*1]や Facebook などのソーシャルネットワークサービス（以下, SNS）も発展し, 発信される情報の量が増えた. 外出先でもより多くの情報を手軽に取得できる便利さは, 移動中に視覚が携帯端末の画面に占有される所謂歩きスマホという社会問題に繋がってしまっている. これは, ユーザーの SNS に対する依存性から生まれる問題であり, これを取り除くのは簡単ではない. そこで本研究では SNS ユーザーに音声を用いた閲覧方法を提案し, 間接的にこの問題を解決する事を目的とする.

数ある SNS の中で, 本研究では発信される情報量の多さと情報の取得の手軽さから Twitter というマイクロブログサービスに焦点を当てる. この Twitter はユーザーどうしがお互いの 140 文字以内の投稿をリアルタイムで共有するサービスで, 文字数上限の短さ故に却って気軽に投稿を行うことができ, 短い時間で様々な情報が得られる利点があり, 2017 年 10 月 27 日の Twitter Japan の発表によると, 国内での月間利用者数は 4500 万人を超えている.

本研究では聴覚を通して SNS ユーザーに情報の内容を提供する. しかし、視覚と違い聴覚は複数の情報を一度に処理する事には向いていない. 視覚では投稿の全体像をざっと見る事ができる為, 効率よく情報の流し読みができる. 一方, 聴覚では投稿内容がすべて読み上げられるまでは, 投稿の全体像を掴む事ができないからだ. 投稿の内容を読み上げる為に Text to Speech(以下, TTS) を用いる. 本研究では, ユーザーが必要な情報を選んで必要な分だけ音声を聴ける様に, 各ユーザーのタイムラインからツイートをトピック別に仕分ける.

今回は歩きスマホをメインの背景としたが, このシステムが実現できれば, 視覚を利用する作業を行いながら Twitter を閲覧するというマルチタスクの実現にも繋がる. 本研究では従来の Twitter におけるトレンド抽出手法が個人のタイムラインにおいてどの程度有効であるかを検証し, 有効な手法を導入した話題選択システムを構築する.

2 ツイートの聴取方法

視覚で Twitter を閲覧した場合, タイムライン全体を流し見する事が出来る為, 話題の把握や情報の取捨選択が容易である. 一方, 聴覚で情報を取得する場合, ツイート一つ一つを聴く必要があり, 話題の把握に時間がかかる. 例えば, 「勝ったーー！」とだけ記されたツイートがあり, いくつかのツイートの後に「ラグビー日本代表がベスト 8 だ！」というツイートがあったとする (図 1). 聴覚だと, 最初のツイート単体だけでは何の話かを理解するのは難しく, 後のツイートを聴いて初めて何が勝ったのかを理解できる. この問題を解決する為に, ツイートを各ユーザーのタイムラインにおけるトレンド別に仕分けて, ユーザーにそのトレンドを提示するという方法を提案する. 使い方のイメージとしては, 音楽の各楽曲がツイートで, それをまとめるプレイリストがトレンドに相当し, ユーザーは希望するプレイリストを選択していくというものである. プレイリストとして提供する為, ツイートを時間内に聴ききれるかも重要である為, ユーザーにはトレンド毎の読み上げ時間も提示する. また, もう一つ問題がある. Twitter はリアルタイムで投稿が増えていくマイクロブログサービスであり, ツイートを聴いている間も新しい投稿は増えていくという点である. 各ユーザーのタイムラインにおけるツイートの時間帯別更新頻度から予想される、新しいツイートの投稿数と読み上げ時間を提示する事で解決を目指す.

Supervisor: Prof. Katunobu Itou
[*1] https://twitter.com

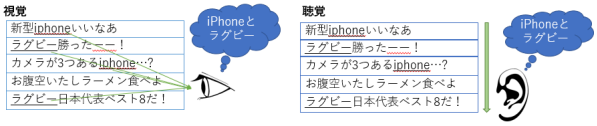

図 1. タイムラインの例

2.1 関連研究

Ghahari らはニュース記事をカテゴリ毎に纏め, カテゴリの一覧をユーザーに提案し, 選択したカテゴリに関連する記事を連続的に再生する様なアーキテクチャである聴覚フローを考案し, 聴覚ブラウジングがアイズフリーな情報取得方法として有効である事を示している [1].

Crammer らは既存の線形分類器とは異なる考え方を導入し, 収束を早めると同時に精度も高くした Confidence-Weighted Algorithm(CW) を提案している [2][3]. 機械学習に入力するデータは, 各特徴の頻度に大きなばらつきがある. また, 低頻度の特徴が大きな影響を持っていることもある. 既存の分類器は各入力データの出現頻度は考慮されていないため, 高頻度の特徴は信頼できる一方, 低頻度の特徴は重みがあまり更新されない為, 信頼度が低いといえる. CW では「過去にどれくらい現れたか」を考慮した上で学習を行う. 重みベクトル w に正規分布 $N(\mu, \Sigma)$ を導入し, 平均と共分散を同時に更新していく. μ はその時点でもっともらしい重みベクトルで, Σ はその重みにどれくらい自信があるのかを表す共分散行列になる. 自然言語処理において既存のアルゴリズムよりも速く高精度である結果を示せたとしている.

辻井らは, マイクロブログにおけるクラスタリングでは, 字数制限によってツイートの特徴が掴み辛く, 重要度の計算において問題が生じるとしている [4]. 一文が短いツイートでは話題の中心となる語が繰り返される事が少なく, 話題の出現頻度で重要度を求める TF-IDF 法が有効でないと述べている. TF-IDF 法とはテキストの特徴を表現する為に, 文章に含まれる単語の重要度を考慮する手法である. TF-IDF の概念は, 珍しい単語が何度も出現する場合, 文章を分類する際にその単語の重要度を上げるというものである. Term Frequency(TF) は単語の出現頻度の事で, 各文書においてその単語がどのくらい出現したのかを意味する.

$$\mathrm{TF} = \frac{\text{文書 A における単語 X の出現頻度}}{\text{文書 A における全単語の出現頻度の和}} \tag{1}$$

Inverse Document Frequency(IDF) は逆文書頻度と呼ばれるもので, 単語が珍しい物なら高い値を, 色々な文書に出現する単語なら低い値を示すものである.

$$\mathrm{IDF} = \log \frac{\text{全文書数}}{\text{単語 X を含む文書数}} \tag{2}$$

この二つを掛け合わせたものを文書の特徴とする. しかしマイクロブログでは, 文書 A において単語 X が何度も出現する事は少なく, 特徴量を求め辛いという事である.

3 システム概要

先ず TwitterAPI を用いてツイートオブジェクトを取得し, ツイートパラメータを参照する事でツイートを取得する. 取得したツイートにどの様な話題があるかを調べる為にクラスタリングを行う. ツイートを分かち書きし, TF-IDF にした物を k-means に入力する. その結果から得られたクラスタを話題グループとし, 各ツイートにラベル付けする. ラベル付けされたツイートを同じラベルで一つのグループにまとめてテキストデータにする, TTS に渡し音声ファイルにする. こうしてできた音声ファイルとそのタイトルとなる話題名をプレイリストの一覧の様にユーザーに提示する. ユーザーは提示された物の中から興味のあるトレンドを選択し, 実際に再生を行う. トレンドの選択には端末の画面を見て行う.

3.1 ツイートの取得

Twitter からのツイートの取得には Twitter 社が開発者向けに公開している API を用いる. この API を用いる事で, Python プログラム内からツイートを取得できるだけでなく, Twitter にアクセスする事や, Twitter アカウントを操作する事が可能になる. 現状, ツイートを取得する為のアカウント認証キーは手作業で調べている為, 各ユーザーが使える様にする為の動作を実装する必要がある.

3.2 形態素解析

形態素解析を用いて取得したツイートを品詞ごとに分類する. 形態素解析には日本語形態素解析システム MeCab を用いる. Python でこれを用いる為に, mecab-python3 というパッケージを利用する[*2]. IPA 辞書は, mecab-ipadic-neologd という辞書プログラムによって作成された物を用いる[*3]. これは Mecab の作者が提供している単語データ[*4]をベースとして, リアルタイムで単語データを更新し続ける特徴を持つプログラムで, 固有名詞の判別精度を向上させている. この辞書を用いて取得したツイートの特徴量を求めていく.

3.3 ツイートの特徴量算出

テキストの特徴量算出には TF-IDF がよく使われる. 関連研究でも触れたが, Twitter で用いる場合に置き換えて説明する. t を単語, d をツイート, $f_{t,d}$ をツイート内で単語が使われている回数とする. この時 TF は,

$$\mathrm{TF} = \frac{\mathrm{f_{t,d}}}{\sum_{t' \in \mathrm{d}} \mathrm{f_{t',d}}} \tag{3}$$

となる. N をツイートの総数, n_t を単語が含まれるツイートの数としたとき, IDF は,

$$\mathrm{IDF} = \log \frac{\mathrm{N}}{1 + \mathrm{n_t}} + 1 \tag{4}$$

となる.

ツイートは最大 140 文字と文章としてはとても短く, 話題の中心となる語が 1 ツイートのの中で繰り返されにくいという問題がある. その為, $f_{t,d}$ の値が殆ど 1 になってしまい, TF が意味をなさない. そこで今回は IDF のみを特徴量とする. IDF を 2 万ツイートで使われる単語に対する重みづけとする. 重みづけで使うタイムライン以外のツイートは, TwitterAPI の Public stream を用いて収集した.

[*2] https://pypi.org/project/mecab-python3/
[*3] https://github.com/neologd/mecab-ipadic-neologd/releases/
[*4] http://taku910.github.io/mecab/#download

3.4 話題の選出

現在のタイムラインから収集したツイート群から話題の抽出を行う．本論では k-meas 法を用いる．k-means 方は決められたクラスタ数でデータを分割し, 最適分割となる分割方法を探索する方法である．クラスタ名は, 各クラスタ毎に形態素解析を行い, 非接続語でない名詞を多い順に並べ, 20 文字以内に収まる分の単語をクラスタ名とする．各ツイートに形態素解析を行った時に, 名詞の頻度がそのままトレンドの抽出に繋がるかを確認した．実験の際にはノイズとなる URL とハッシュタグは取り除いた．実験には最新の 200 ツイートを用いた．以下は結果の一部抜粋である．名詞と分類された頻度の高

表 1. 実験結果

名詞	C	本	中	雨	モンスター	機体	新刊	研究	歴史	冬	久川
回数	21	18	14	11	11	11	11	10	9	7	7

いものはそれ単体では何を指すのか不明であり, 何を意味するのか分かる名詞の頻度は横ばいに並んでいた．その為, ただ形態素解析をしただけでは話題名の抽出は難しい．複数の名詞を提示する事でユーザーにどの様なクラスタなのかを想像してもらう．ここで使われる単語は, 特徴量算出で用いられる品詞と同様である．

3.5 音源の用意と再生時間の提示

ツイートの読み上げには IBM Watson の TTS を用いる[*5]．TTS は Microsoft など様々な所から提供されているが, IBM Watson は無料試用の範囲内で出来る事が多い為採用した．読み上げの音源は Watson の API を通してツイートのテキストデータを TTS に渡し, wav ファイルを生成させる事で用意する．再生時間はこのファイルのパラメータを参照してユーザーに提示する．

3.6 システムの改善案

まず各ユーザーが利用できる為のプログラムを実装する．これは Twitter の既存のサービスであるアプリケーション認証という動作を用いる．アプリケーション認証とは, アカウントのログイン ID とパスワードを入力する事でアプリケーションを使用可能になるというサービスである．このサービスを使用する事で任意のアカウントの API を取得しタイムラインを読み込むことが可能になる．

現状のシステムは Twitter の利便性の一つであるリアルタイムな更新に対応していない．過去のツイートを聴取している間にも新しいツイートは増えていく．その為, 聴取している間にどの程度ツイートが増えるのかをユーザーに明示する必要がある．しかし, タイムラインが流れる速度はユーザーによって異なる為, 一般化し辛いと考えられる．アプリケーションとしての即時性に欠けるが, 各ユーザーのタイムラインの速度を逐一観察し, その結果に基づいて予想されるツイート数を提案するというものが考えられる．

アプリケーションを実装する為に, ユーザーインターフェース (UI) を決めていく必要がある．Twitter の利用者が好きな時に気になるトレンドについてのツイートを音声で聴ける様な UI が求められる．イメージとして近いのは音楽アプリでプレイリストを選ぶ感覚である．

[*5] https://cloud.ibm.com/apidocs/text-to-speech

4 評価実験

普段から Twitter を利用している 3 名を対象に実験を行った．3 名とも 20 代男性で, フォロー数の平均は 592.33, 分散は 6871.89, 標準偏差は 92.89 である．

実験データは被験者のタイムラインから 18 時時点での最新 400 ツイートを扱った．これはフォロー数 500 人ぐらいのタイムラインだと 400 ツイートで 4 時間から 6 時間分の物になるからである．

クラスタ数は 5, 10, 30, 50 クラスの 4 種類に設定した．400 ツイートのタイムラインを 5 種類集め, 手作業で話題数を数えたところ, 平均 74.4, 分散 65.84, 標準偏差 8.11 になった．この事からクラスタ数は 70 ぐらいまでが適切であるが, 提示する話題数としては多すぎる為, 上限を 50 とし, 満遍なくクラスタ数の種類を分ける為に設定したのが上記の 4 種類である．

4.1 クラスタリングの評価

クラスタリング結果の評価は, 次に示す正規化相互情報量 (normalized mutual information) を用いて行う．

$$\mathrm{NMI}(\mathrm{C},\mathrm{T}) = \frac{\mathrm{MI}(\mathrm{C},\mathrm{T})}{\max(\mathrm{H}(\mathrm{C}),\mathrm{H}(\mathrm{T}))} \tag{5}$$

C は生成されたクラスタ集合, T は正解クラスタ集合であり, MI は相互情報量, H はエントロピーを表す．NMI は, 0 から 1 の間を取り, 値が大きい程生成されたクラスタが正解クラスタ集合に類似していることを示す．

4.2 興味度評価

ユーザーが選択した話題にどれだけ興味のあるツイートが含まれているかを評価した．まず提示した中から話題を選択してもらう．その後, タイムライン全てのツイートに対して 5 段階評価で興味度の評価をしてもらった．その後, クラス数と選択されたか否かによる違いを観察した．5 段階評価の内容は次の通りである．

1. 興味がない
2. それほど興味はない
3. どちらとも言えない
4. まあまあ興味がある
5. 興味がある

5 考察

5.1 クラスタリングの評価

表 2. k=5

Timeline	TF-IDF	IDF
A's	0.219	0.276
B's	0.193	0.264
C's	0.208	0.263

表 3. k=10

Timeline	TF-IDF	IDF
A's	0.287	0.378
B's	0.224	0.352
C's	0.256	0.364

表 4. k=30

Timeline	TF-IDF	IDF
A's	0.332	0.401
B's	0.295	0.383
C's	0.301	0.39

表 5. k=50

Timeline	TF-IDF	IDF
A's	0.416	0.451
B's	0.402	0.497
C's	0.431	0.501

どのタイムラインと k の値に関わらず, TF-IDF を利用する場合より IDF のみを利用したクラスタリングの方が良い結果になった．このことから, IDF のみを用い

たクラスタリングは有効であるといえる．ただし，k=5, 10 の時は性能が低く，k=30, 50 の時も決して高い値とは言えない．今回はクラスタ名を複数の単語で表す為，k=30, 50 ぐらいの結果ならば，クラスタ名から期待できる内容のツイートをある程度取得できると考えられる．

5.2 興味度評価

結果を見ると，選択された話題群の中に興味度の高いツイートが集まる傾向になった．この事からユーザーに興味を持たせるツイートを一つのクラスタに纏める事が出来ているといえる．

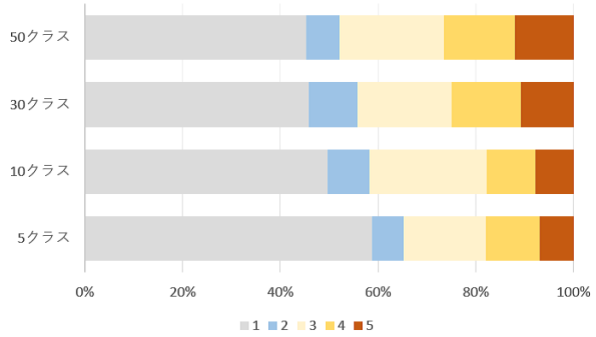

図 2. ツイート興味度内訳:選択された話題群

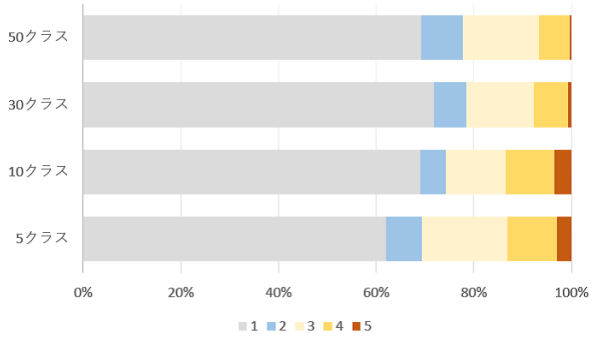

図 3. ツイート興味度内訳:選択された話題群

問題点として，提示する話題名に，動詞の連用形や，形態素解析の性能面の問題で本来名詞でない単語が含まれることがある．その様な単語が多い話題名は，ユーザーに内容の推定を促し辛く，興味のある話題として選択され辛いという物がある．次の表はは名詞以外の単語が含まれた例である．これは 3 ツイートしかないクラスタで

表 6. 名詞以外の単語が含まれたクラスタ名

単語	ー	中	エヴァンゲリオンマンチョコ	口	休み
回数	2	2	1	1	1

起こった事で，「寝れへんのにー！」と，「夜更かししちゃおー」いうフレーズから伸ばし棒が取得された．形態素解析をすると次のようになる．

1. 寝れ/へん/のに/ー/！
2. 夜更かし/し/ちゃ/お/ー

「のに」は接続助詞で，「お」は接頭詞であり，後には名詞が続くという様な判定を行った為，名詞と判定されてしまったと考えられる．選択されなかった話題の中で興味度 4, 5 のツイートの内，正しく形態素解析されなかったツイートは 41.2% あった．

選択されなかった話題に含まれる興味度の高いツイートを選択されるにはどうすれば良いのかを考察する為に，興味度の高いものと低いもので分けて観察をしたところ，文字数に大きな差がある事が分かった．この事から漏れてしまったツイートを拾う基準になるのではないかと考えられる．使い方としては，クラスタを読み上げる順番を決める時に，文字数を基準にしたと仮定して，一通り再生した後に選ばれなかった話題から再生するか選択できるようにするという物が考えられる．

表 7. 選択されなかったツイートの文字数

興味度	平均	分散	標準偏差
1, 2	26.84	534.12	23.11
4, 5	118.25	1469.02	38.33

また，話題名として使われている単語が被る事が 6% あった．被験者からは「気にならなかった」という回答を得られた為，今回は問題なしとする．

提案手法だと興味のないツイートを除外する事ができないという問題点もあり，これを解決できるとより良いシステムになる．また，被験者から「50 クラスは文字が多すぎる」という指摘を頂いた．今回は 1 クラス 20 文字で話題を提示しているので，1000 文字をユーザーに読ませていた事になる．その為，今回の手法の場合，クラス数は多くても 30 程度にする必要がある．

6 あとがき

今回の研究では，音声ブラウジングにかける時間を出来るだけ減らす事を目的として，ユーザーの興味度が高いツイートを提示できる様にする為に，IDF を特徴量としたツイートのクラスタリングと，そのクラスタ名の提示を行った．その結果，いくつかのクラスタに興味度の高いツイートを集める事が出来，興味度の高いツイートが集まったクラスタをユーザーに選択させる事ができた．特に今回は特徴量が結果に反映されやすい k-means 法を利用しているため，IDF のみを特徴量とする手法が有効であると言える．

参考文献

[1] Romisa Rohani Ghahari, Davide Bolchini, “ANFORA: Investigating Aural Navigation Flows On Rich Architectures”, 13th IEEE International Symposium on Web Systems Evolution (WSE) (2011)

[2] Mark Dredze, Koby Crammer, Fernando Pereira, “Confidence-Weighted Linear Classification”, International Conference on Machine Learning (ICML) (2008)

[3] Mark Dredze, Koby Crammer, Fernando Pereira, “Confidence-Weighted Linear Classification for Text Categorization”, Journal of Machine Learning Research 13 (2012)

[4] 辻井由佳, 西山裕之, “マイクロブログの特徴を考慮した文書クラスタリング手法の提案と実装”, 人工知能学会誌 Vol. 27 No. 6 (2012)

[5] Hao Cheng, Kien A. Hua, Khanh Vu, ” Constrained locally weighted clustering”, Proc. of the VLDB Endowment, Vol.1, No.1, 2008