Apache Spark を用いた異種データ解析システム

川森　善紀†　井上　潮‡

†東京電機大学 工学研究科 情報通信工学専攻 〒120-8551 東京都足立区千住旭町 5 番

‡東京電機大学 工学部 情報通信工学科 〒120-8551 東京都足立区千住旭町 5 番

E-mail: †16kmc10@ms.dendai.ac.jp, ‡inoueu@mail.dendai.ac.jp

あらまし 近年、ビッグデータ解析により得られた結果を経営戦略などに役立てる需要が高まっている。最近ではビッグデータとしての Twitter サービスから得られる Tweet をストリームデータとして集め、抽出した特徴を用いて別種類のデータを予測する材料として活用する事例も多い。Tweet 処理する際には、想定外の語句に対するデータ再取得や、Tweet から特徴を抽出する処理など、新たな結果を取得するまでに多くの時間を要する。そこで、本研究では分散処理フレームワークの一つである Apache Spark を利用して、高速に Tweet の抽出・加工を行い、ユーザの持つデータと組み合わせた解析用データセットの生成及び解析を行えるシステムを構築した。

キーワード Apache Spark, Twitter, ビッグデータ

1. はじめに

近年、収集したビッグデータを解析して得られた知見を様々な方法で活用する取り組みが行われている[1]。その中でも、Tweet や商品レビューなどの短い文章から抽出した情報を用いて、商品、サービス、株価などの時系列データを解析する研究が多くなされている。

Tweet に対し解析を行う研究では Tweet に対して予めキーワードを設定し、フィルタリングを行った結果を収集している。この場合、キーワードの変更が必要になると、Tweet を新たに収集し直す必要がある。更に、AIP 制限や実行速度の関係から実行回数が制限されている場合も多い。さらに、収集した Tweet を用いて解析する対象となるデータも限定されている場合が多い。

本研究では、可変的にかつ高速に Tweet をフィルタリングするために、分散処理フレームワークである Apache Spark[3]に着目した。

Apache Spark は RDD と呼ばれる分散データセットに対して、演算をメインメモリ上で行なうことにより、ディスク IO を削減した高速処理が行える。更に、SQL を用いたデータ処理が可能な Spark SQL、ストリーム処理を実現する Spark Streaming、機械学習ライブラリである MLlib, グラフ処理ライブラリである GraphX と言ったライブラリを含んでおり、機械学習や視覚化などを手軽に実装することができる。

このような分散処理フレームワークを利用し、蓄積した Tweet に対して分散処理を用いて高速に抽出を行う。また、抽出した Tweet に対して分析を行い、ユーザの持つ株価のような異種のデータを組み合わせたデータセットの作成及び解析をブラウザ上で行うことのできるシステムの開発を行った。

2. 関連研究

駒田ら[2]は、Twitter データを用いて係り受け解析を行う際に、Tweet に含まれる表現に適合するパターンから、フィルタリングと係り受け解析を行い、関連度を計算することで、未知である属性語の同定を複数回行ない、Twitter に投稿された商品評価 Tweet からの良い属性語の自動抽出を実現した。

大部ら[4]は、ソーシャルメディアを用いて学園祭などのイベント情報の収集および分析を支援するシステムの実現を目的とし、収集した Tweet テキストへの強調、着色やバブルチャート、ワードクラウドを地図情報とともに出力することにより、ユーザがイベント情報を発見することを支援するシステムを開発した。

佐藤ら[5]は、インターネット上のテキストデータの解析結果と株価に相関性があるかを確認することを目的として、Tweet と Web ニュースのデータに形態素解析を行いその結果と株価の増減の相関係数を算出した。

村上ら[6]は、ユーザが設定した検索語句を用いて、その単語及び関連語句のポジティブ、ネガティブを判定するシステムをスマートフォンアプリケーションとして開発している。

加藤ら[7]は、Apache Spark を利用した機械学習において、パーティション数とノード数を変化させることで、実行時間の測定を行っている。この研究では多く処理を割り当てられたノードの律速の影響で処理時間が遅くなることが確認されている。そのため、大量にジョブが生成される処理を行う際にタスクが公平に配分されないことが課題として挙げられている。

3. 課題設定とアプローチ

3.1 文章データの可変解析での問題点

Tweet のような文章において、複数のキーワードを用いて連続的に検索、解析を行う場合、対応するデータを取得する方法として、Twitter では Twitter API search/tweets[8]がある。これは、解析対象となる物のキーワードを入力し取得を行なうが、API 制限によりリクエスト回数が 15 分あたり 15 回に制限されているため、複数の検索ワードを用いた場合や、解析段階での別キーワードでの再検索をするにはコストが大きい。このような問題を解決するためには、キーワードを指定せずにデータを取得する必要があるが、取得したデータが膨大になるため、フィルタリングに大きな時間がかかる。

3.2 異種データ解析における問題点

Tweet のような文章データを用いて株価のような異種のデータを解析するためには、Tweet を感情分析などの方法を利用し数値化し、ペアとなるデータとつなぎ合わせ、解析エンジンへ入力する必要がある。これらの手順をすべて手動で複数回解析を行うような環境ではユーザの負担が大きい。

3.3 アプローチ

本研究では、高速なデータ抽出を実現するために、キーワードを指定せずに収集した Tweet を Apache Spark によって高速なフィルタリングを実現し、3.1 の問題について解決を図る。また、抽出したデータの加工、及びユーザの持つデータとの連結及び解析を一連のシステムとして実装することで、ユーザによる操作を削減し、3.2 の問題について解決を図る。

4. システム実装

4.1 システム処理手順

システム構成を図 1 に示す。以下に本システムが実装する処理の手順を示す。

1. データ収集

Twitter Streaming API を用いてつぶやかれたすべてのデータから日本語のもののみを収集する。本システムでは常に収集を行っている。

2. 抽出及び数値化

Apache Spark を用いた分散処理によって、ユーザが指定したキーワードに関連するデータを抽出し、数値化を行う。

3. ジョイン処理

収集した Tweet に対して抽出及び数値化を行ったデータとユーザが入力したデータを結合し、解析用データセットを生成する。

4. 解析処理

作成したデータセットに対してランダムフォレストを用いて解析を行い、学習モデルの生成及び交差検証を用いたテストを行う。

また、2 及び 4 の処理を Apache Spark を用いた分散処理で実装することによって抽出速度の高速化を図る。次に、以上の機能をブラウザ上で実行を可能にすることでユーザ負担の軽減を図る。対応データについては常に収集と蓄積を行うストリームデータであり、安定的にデータを取得できる Tweet データに固定し、ユーザ入力データを変更可能なシステムとした。

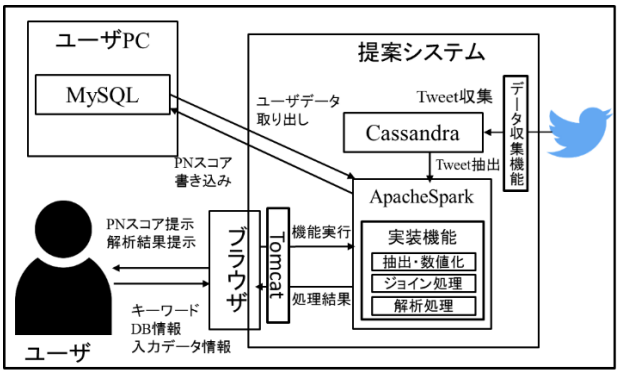


図 1 システム概要

4.2 解析概要

本システムは、ユーザが指定したキーワードに関連する Tweet を抽出し、Tweet がキーワードに対してポジティブもしくはネガティブのどちらの属性を持つか判定を行う感情極性分析を行う。そして、分析を行った結果から算出した PN 判定スコアと、ユーザが入力したデータを組み合わせて解析を行うことである。PN 判定を行う際の前処理として、日本語形態素解析ライブラリの Kuromoji[9]を用いて形態素解析を行った。また、算出した PN スコアとユーザが入力したデータを組み合わせたデータの解析には、Spark ML ライブラリ内のランダムフォレスト分析を利用した。

4.3 数値化手法

数値化には、日本語評価極性辞書（名詞編）[10]を参照し、一致する単語に対してネガポジ判定を行った。日本語評価極性辞書には、各単語に対して P,N,E と言う形で極性が決められており、それぞれポジティブ、ネガティブ、どちらでもないという意味となる。ここから、図 2 のように p を 1 ポイント、n を-1 ポイント、e を 0 ポイントとして各 Tweet を形態素解析した結果から辞書に対応する単語の極性値を足し込む事によってスコアを集計

する。

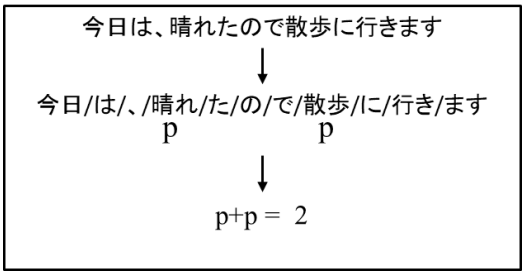


図 2 PN スコア集計例

4.4 ジョイン処理

ジョイン処理は、ユーザが入力したデータを算出した PN スコアと結合し一つのデータセットとする処理である。本システムでは、図 3 のように日付、Tweet に対して、抽出及び数値化を行い算出した PN スコア、終値と、終値の上昇下降を正解データとして記したものをジョイン操作によって一つのデータにまとめ、データセットを作成している。

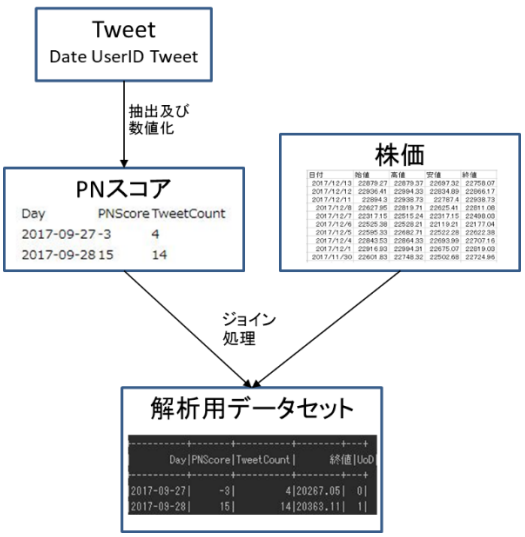


図 3 解析用データセット作成の流れ

4.5 システム動作

本システムではユーザの入力した情報から関連情報の抽出、PN スコアの算出及びプレビュー、PN スコアのデータベースへの書き込み、PN スコアとユーザ入力データを組み合わせたランダムフォレスト解析を行う。各機能について順を追って説明を行う。なお、Tweet の利用方法については感情極性分析を行う。辞書を読み取り、単語に対応する数値を利用する形式であるため、ユーザ定義の辞書で感情極性分析以外の分析方法にも対応可能である。しかしシステム自体の煩雑さや辞書の用意など、ユーザの負担を回避するため今回は機能としての実装を行っていない。以下、キーワードを「日経」とし、ユーザ入力データには図 4 の日経平均株価を入力した場合の動作を説明する。

日付	始値	高値	安値	終値
2017/12/13	22879.27	22879.37	22697.32	22758.07
2017/12/12	22936.41	22994.33	22834.89	22866.17
2017/12/11	22894.3	22938.73	22787.4	22938.73
2017/12/8	22627.95	22819.71	22625.41	22811.08
2017/12/7	22317.15	22515.24	22317.15	22498.03
2017/12/6	22525.38	22528.21	22119.21	22177.04
2017/12/5	22595.33	22682.71	22522.28	22622.38
2017/12/4	22843.53	22864.33	22693.99	22707.16
2017/12/1	22916.93	22994.31	22675.07	22819.03
2017/11/30	22601.83	22748.32	22502.68	22724.96

図 4 入力データ（日経平均株価）

初めに、ユーザは図 5 のメインページに検索ワード、PN スコア書き込み用のデータベース情報、入力データの格納されたデータベース情報、入力データの日付と予測を行う情報となるカラム名を入力する。ユーザ入力データについては、日付が yyyy-mm-dd の形式であり、予測対象が数値形式の時系列データであればどのようなデータでも解析を行うことができる。

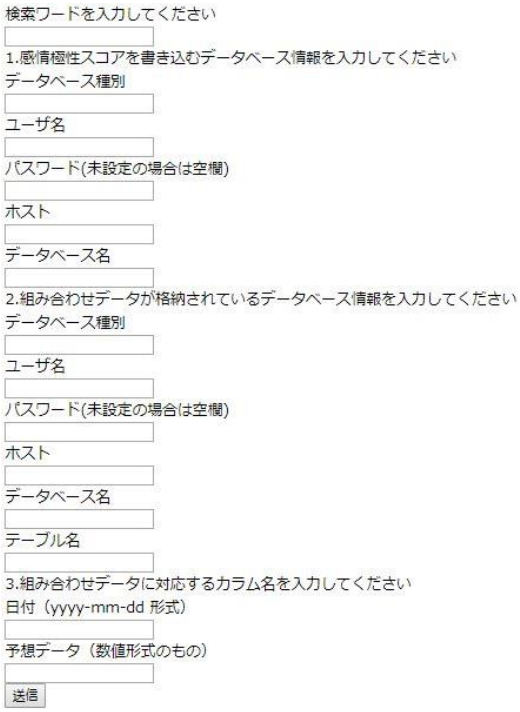


図 5 メインページ

関連 Tweet の抽出が完了すると図 6 のページに移動する。このページは算出した PN スコア、日付、日付ごとに抽出された Tweet 数のプレビューを確認することができる。さらに、以降の解析や PN スコアデータをユーザ自身が使い慣れた環境で行うことを想定し、Download ボタンを押すことで CSV 形式のデータをダウンロードすることができる。書き込みボタンでは、ユーザが図 5 の 1.に入力したデータベース情報を元に PN スコアをデータベースへと書き込む。作成されるテーブル名は「キーワード:PNScore」という形式となり、同名のデーブルがある場合は削除を行ってから新たにテーブルの作成を行う。

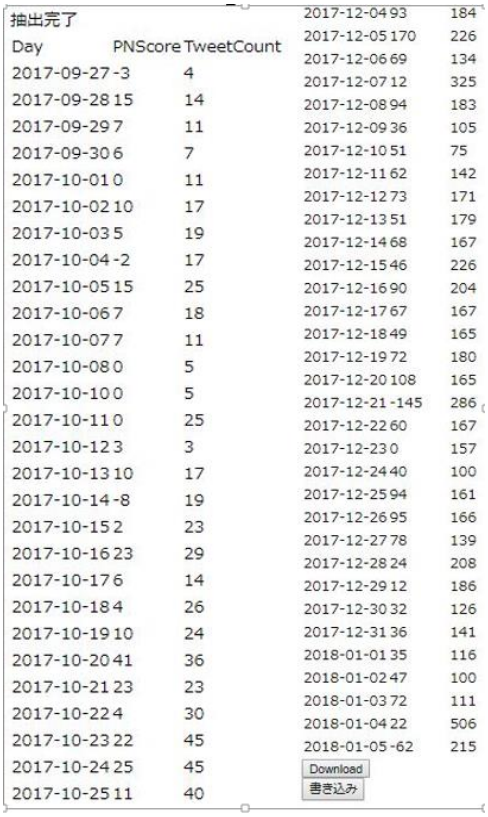


図 6 PN スコアプレビューページ

データベースへの書き込みが完了すると、図 7 のページに移動する。このページでは PN スコアのデータベースへの書き込みが完了したことを通知するページであり、解析を必要としないユーザはタブを閉じて終了する。解析を行う場合は解析開始ボタンを押すことで図 8 のようなデータセットが作成され、解析が行われる。

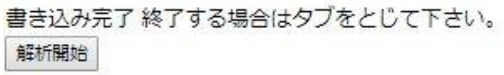
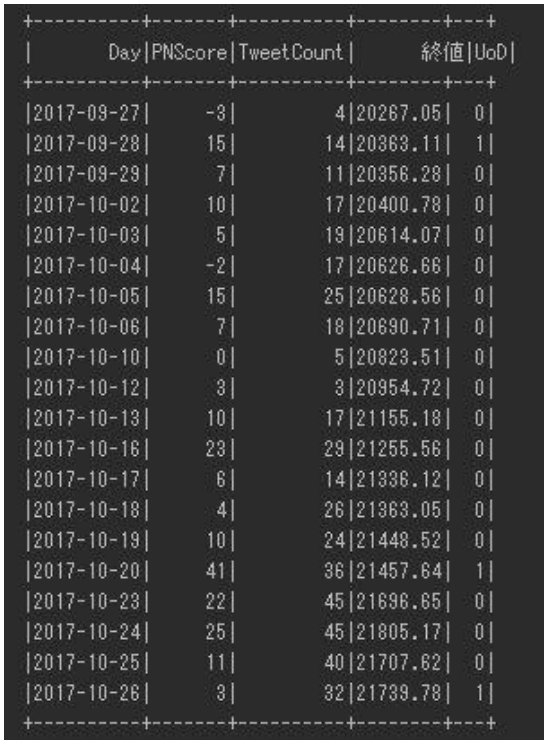


図 7 データベース書き込み完了ページ



Day	PNScore	TweetCount	終値	UoD
2017-09-27	-3	4	20267.05	0
2017-09-28	15	14	20363.11	1
2017-09-29	7	11	20356.28	0
2017-10-02	10	17	20400.78	0
2017-10-03	5	19	20614.07	0
2017-10-04	-2	17	20626.66	0
2017-10-05	15	25	20628.56	0
2017-10-06	7	18	20690.71	0
2017-10-10	0	5	20823.51	0
2017-10-12	3	3	20954.72	0
2017-10-13	10	17	21155.18	0
2017-10-16	23	29	21255.56	0
2017-10-17	6	14	21336.12	0
2017-10-18	4	26	21363.05	0
2017-10-19	10	24	21448.52	0
2017-10-20	41	36	21457.64	1
2017-10-23	22	45	21696.65	0
2017-10-24	25	45	21805.17	0
2017-10-25	11	40	21707.62	0
2017-10-26	3	32	21739.78	1

図 8 解析入力データセット

このデータセットは、日付、PN スコア、Tweet 総数、日経平均株価終値、増減で構成されている。UoD は増減を意味し、ユーザの指定した予測データから、次の日数値が上昇している場合は 0、下降している場合は 1 を割り振り、正解データとしてデータセットに付け加えている。解析が完了すると図 9 のページに移動する。このページでは解析に利用したカラムリスト、出力された木構造、交差検証方を用いた分析モデルの評価詳細の確認と各項目をテキスト形式でダウンロードすることができる。

図 9 解析結果提示ページ

5. 評価

本システムは、蓄積した Tweet データからキーワードに関連する Tweet の抽出及び解析を行う処理を様々なキーワードを用いて複数回実行することを想定している。そのため、システムの中で行われる抽出及び数値化、書き込み、解析の 3 つの処理を合計したシステム実行時間に対して、分散を行わないワーカーノード 1 の場合と、ワーカーノードが複数ある場合での実行時間を比較する。また、システム全体として、ユーザ負担軽減効果を確認する。

5.1 実行速度評価

評価に用いた PC、フレームワーク、データベースの詳細を表 1 に示す。

表 1 評価利用 PC 等の詳細

OS	Linux CentOS 7 64bit, RAM 4.00GB
プロセッサ	Intel(R) Core(TM) i5-2400S 2.50GHz
RDB	MariaDB 10.1.29
NoSQL	Apache Cassandra 3.0 (DSC30)
分散処理フレームワーク	Apache Spark2.2.0
Tweet総数	40,192,978件
Tweet収集期間	2017年9月27日~2018年1月5日

実行時間の測定には Apache Spark のワーカーノード数を 1~5 まで変化させ、それぞれ 5 回測定を行い、平均値を算出した。その結果を図 10 に示す。

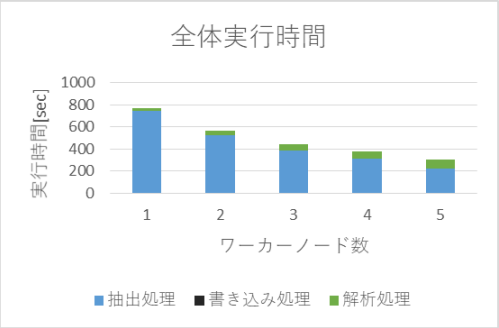


図 10 抽出処理実行時間

従来のシステムではユーザによるキーワードの決定後にデータ収集が行われている。そのため、API 制限により収集できるデータ量に限界があることや、キーワードの変更を行う場合データの再収集が必要である場合がある。また、村上らの研究では、PN 判定を API によって行っていたため 100 回/日の実行回数制限があった。一方、本システムではすべてのデータを蓄積しそれを分散し高速に処理することで、図 10 に示すようにデータ抽出速度を向上させることができ、短時間で複数のキーワードを用いたデータセットの作成が可能になった。しかし、解析処理については分散化を行うことで実行速度が悪化している。これは、ノードの処理待ちによって結果集計に時間がかかり、速度が悪化しているものと思われる。その為、パーティションやジョブの最大数をチューニングし、最適な設定について検討する必要がある。

5.2 全体評価

本研究では、ユーザ負担を目的として Tweet 抽出からユーザ入力データとの解析までをシステム化した。従来は、Tweet 抽出、ユーザデータとの一体化、解析エンジンを用いた解析とそれぞれを個々に行う必要や操作に応じたプログラムの作成を行う必要があった。本システムではユーザによる操作はキーワード、データベース情報、及び入力データ情報の入力と、PN 解析、PN スコア書き込み、ランダムフォレスト解析とそれぞれの操作を実行するボタンクリックのみである。そのため、ツールの切り替えやプログラムの記述が必要ないため、ユーザ操作を削減し、負担を軽減し、複数のキーワードでデータセットを作成することができた。しかし、ランダムフォレスト解析において交差検証法の精度が 1.0 であり、精度が良すぎることから過学習やサンプルデータの不足などの原因が考えられる。

6. おわりに

本稿では、蓄積した Tweet から繰り返し PN スコアの抽出が行える機能と、抽出したデータとユーザが入力したデータをあわせて解析を行う機能を持ったシステムの提案と実装及び評価について述べた。

本システムでは、利用できるデータベースが MySQL のみであったため、他のデータベース製品への対応と、Tweet に対してのリアルタイムなストリームデータ解析の実装や、

解析精度の向上、PN 判定を日本語以外でも行えるようにすることが今後の課題として挙げられる。

参 考 文 献

[1]総務省,平成 27 年度版情報通信白書第 2 部,ICT が拓く未来社会
http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc254320.html

[2] 駒田康孝,山名早人, 商品評価ツイートからの属性語自動抽出手法の提案, DEIM Forum 2014 B5-6

[3]Apache Spark Lightning-fast cluster computing
https://spark.apache.org

[4] 大部達也忠親,新谷 虎松大囿,"ツイートの可視化によるイベント情報分析支援システムの試作" ,IEICE Technical Report(2015)

[5] 佐藤謙太,小高知宏,黒岩丈介,白井治彦,”ネガポジ解析による Web データと株価変動の相関関係評価”, 福井大学大学院工学研究科研究報告第 63 巻 2015 年 3 月,pp75-86

[6]村上奈緒,尼岡利崇, Twitter 上で任意の検索語句に対するネガポジ度を判定し可視化するアプリケーションの開発と研究, エンタテインメントコンピューティングシンポジウム 2014 論文集,pp261 - 265

[7] 加藤香澄,竹房あつ子,中田秀基,小口正人, 大規模データ分散処理プラットフォーム Apache Spark を用いた分散並列機械学習に関する考察, DEIM Forum 2017 H4-3

[8] Standard Serche API
https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

[9]kuromoji
https://www.atilika.com/ja/kuromoji/

[10] 東山昌彦, 乾健太郎, 松本裕治, 述語の選択選好性に着目した名詞評価極性の獲得, 言語処理学会第 14 回年次大会論文集, pp.584-587, 2008.