

マイクロブログ分析のための因果モデル作成支援システムの試作 On a Support System for Building Causal Models to Analyze Microblogs

片山 真也[†] 白松 俊[†] 大園 忠親[†] 新谷 虎松[†]
Shinya Katayama Shun Shiramatsu Tadachika Ozono Toramatsu Shintani

1. はじめに

本研究では、ユーザが想定する因果モデルに従って、マイクロブログの監視、予測をするためのシステムを実現することを目指している。本稿では、因果モデルに従って、マイクロブログを監視するシステムについて述べる。因果モデルは、ある特定のキーワードが出現したということを示すノードと、それらの時間的関係を表すアークによって表現される有向グラフである。例えば、「インフルエンザ」が流行した後に、「マスク」の販売量が増加する場合には、「インフルエンザ」ノードから、「マスク」ノードに向かって、アークを引くことで、関連性を示すことができる。

マイクロブログは、従来のブログと比較して、情報の即時性と、規模の面で上回っていると考えられる。これらに着目した先行研究のうち、Sakaki ら[1]は、マイクロブログのひとつである Twitter を対象に、投稿(ツイート)された文章から、特定の事象の発生を検知するアルゴリズムを提案している。また、荒牧ら[2]はツイートから、インフルエンザの流行を予測したと報告している。先行研究では、発生済みの事象をツイートから発見したが、本研究では、将来発生し得る事象の獲得を目指し、Twitter を利用して、因果モデルに従ったパターンを監視することを目指している。

本稿では、因果モデル作成の支援、および、因果モデルに従ったツイートパターンの監視を半自動的にこなす Agent の制御をするためのシステムについて提案する。

2. システムの概略

2.1 システムの構成

本システムの構成図を図 1 に示す。本システムにより、ユーザは、ツイートの収集、検索をおこない、抽出した情報を、因果モデルの作成に活用する。また、本システムは因果モデルに従ってツイートパターンの監視、発見もおこなう。

Streaming Agent は、Twitter の Streaming API にリクエストを送信し、ツイートの情報が格納された JSON 形式のデータを受け取る。そして、受け取ったツイートのデータと、Agent 自身の ID を出力として、前処理モジュールに渡す。

Search Agent は、Twitter の Search API にリクエストを送信し、入力された検索語にマッチしたツイートの情報が格納された、JSON 形式のデータを受け取る。このデータも、Streaming Agent と同様に、ツイートデータと、Agent の ID を出力として、前処理モジュールに渡す。上記 2 種類の Agent は複数生成することができ、それぞれ異なった条件の下で、API にリクエストを送信する。

ユーザ情報取得 Agent は、REST API にリクエストを送信し、Search API からは収集することができない Twitter ユーザの情報を取得する。そして、それをを用いて、Search

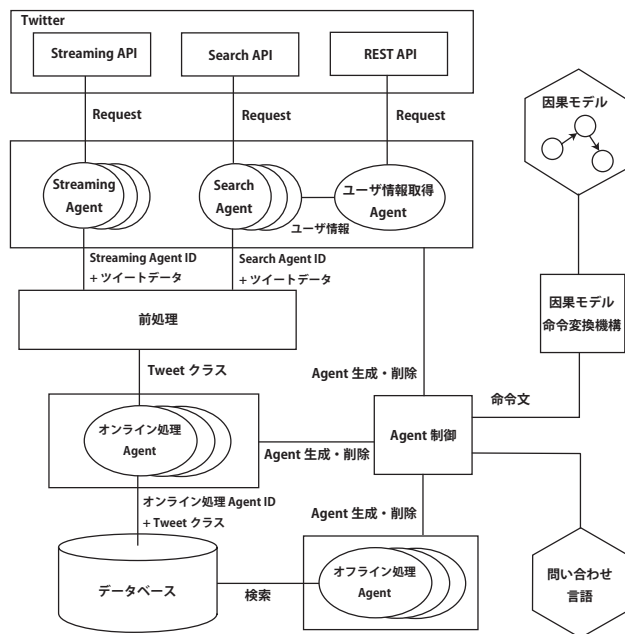


図 1 システム構成図

Agent が受け取ったツイートのデータを補完する。

前処理モジュールでは、前述の Agent から受け取ったデータを、扱いやすい形式に変換する。その形式を Tweet クラスと呼ぶ。Tweet クラスのプロパティを表 1 に示す。

オンライン処理 Agent は、前処理が完了し、データベースに保存可能な形式となったツイートのデータを、データベースへと保存する前に、あらかじめ指定された条件の下で監視する Agent である。この Agent は、ツイートデータが条件にマッチした場合、条件にマッチしたツイートデータを、自身の ID とともに、データベースへと保存する。また、因果モデルを入力として与え、ツイートがそれに従ったパターンで現れるかどうかを観察することも可能である。オンライン処理 Agent も、前述の Streaming Agent, Search Agent と同様に、複数生成することができる。

オフライン処理 Agent は、データベースに格納されたデータに対し、バッチ処理の形式で検索を行う Agent である。検索については、Twitter 特有の条件を検索文で指定することができる。例えば、とあるユーザのフォロワーのみを対象とした検索をおこなうことが可能である。

因果モデル命令変換機構は、因果モデルを入力として受け取り、Agent 制御モジュールが処理することのできる命令文に変換する。変換して生成された命令文は、Agent 制御モジュールに渡され、Agent に、因果モデルに従ったツイートパターンを監視させる際に使用する。

Agent 制御モジュールは、前述した、各 Agent の生成と削除をおこなう。Agent を生成する際には、その Agent がどのような動作をするのかを指定する。

[†]名古屋工業大学 Nagoya Institute of Technology

表 1 Tweet クラスのプロパティ

status_id	ツイートの ID
text	ツイートの本文
user_id	ツイートをしたユーザの ID
screen_name	ツイートをしたユーザの名前
in_reply_to_status_id	リプライ先ツイートの ID
favorited	お気に入り登録されているか否か
retweet_count	ツイートがリツイートされた回数
created_at	ツイートされた日時
geo	ツイートに埋め込まれた位置情報
media_urls	ツイート中のメディア(画像, 動画)の URL
urls	ツイート中の URL
hashtags	ツイート中のハッシュタグ
morpheme	ツイート本文の形態素解析結果

本システムにより、ユーザは、Twitter から収集したツイートデータから、関連性の強い事象同士を探し出し、それを、因果モデル作成に活用できる。また、本システムは因果モデルに沿ったツイートパターンを監視、発見する。

2.2 Tweet クラス

図 1 に示した Tweet クラスは、Streaming API または、Search API から取得した、JSON 形式のツイートデータから、必要な情報を取り出し、扱いやすい形式に変換したものである。実装言語には Java を用い、JSON 形式のツイートは、JSONIC を用いて解析した。Tweet クラスのプロパティを表 1 に示す。

プロパティ `morpheme` は前処理モジュールによって追加されるデータであり、形態素解析器の MeCab を用いて、ツイートの本文を形態素解析した結果が格納される。

`urls` には、ツイート中の URL が格納されるが、URL が短縮されていた場合には展開をし、元の URL を格納する。

3. 問い合わせ言語

本稿では、ユーザが既存のデータから有益な情報を得ることにより、因果モデルの作成に活用できることを目指し、Twitter に特化した問い合わせ言語を構築した。問い合わせ言語の抜粋を図 2 に示す。

(1)は Streaming Agent を生成する問い合わせである。`agentname` には生成する Agent の名前を指定する。`datetimefrom, datetimeteto` はそれぞれ、Agent が API からツイートの収集を開始するタイミング、終了するタイミングである。

(2)は Search Agent を生成する問い合わせである。`agentname, datetimefrom` および、`datetimeteto` に関しては、(1)と同様である。`query` は Search API に入力する検索語であり、`schedule` にはツイートの収集のスケジュールを指定する。例えば、`schedule` には、週に一度だけ収集をおこなうといったスケジュールを指定できる。

(3)はオンライン処理 Agent を生成する問い合わせである。`rulename` には Agent の名前を指定する。この問い合わせでは、`condition` の条件を満たすツイートに対して、`statement`

```
(1) CREATE STREAMING AGENT agentname
    (datetimefrom, datetimeteto);

(2) CREATE SEARCH AGENT agentname
    (query, datetimefrom, datetimeteto, schedule);

(3) CREATE RULE rulename
    IF condition THEN statement;

(4) CREATE RULE rulename
    APPLY MODEL modelname;

(5) CREATE QUERY queryname
    SCHEDULE schedule {select};

(6) SELECT variables FROM graphs WHERE {condition};
```

図 2 問い合わせ言語の抜粋

に記述された処理をおこなう Agent を生成することができる。

(4)は(3)と同様に、オンライン処理 Agent を生成する問い合わせであるが、この問い合わせでは、因果モデルを Agent に与え、ツイートのパターンがそれに従っているか否かを監視させることができる。

(5)はオフライン処理 Agent を生成する問い合わせである。`queryname` には、Agent の名前を指定し、`schedule` には、検索のスケジュールを指定する。`select` には(6)に示すような、データベースに対する検索式を記述する。検索式には、SPARQL ライクな記述を使用することができる。

以上の問い合わせ言語を使用することで、Twitter の Search API でのツイート検索と比較して、詳細な検索をおこなうことができる。例えば、「原発デモ」という検索語を含むツイートをしたユーザの、フォロワーのツイートを検索するといったことが可能である。この例では、「原発デモ」についてのツイートを見た可能性が高いユーザのツイートを観察することで、「原発デモ」という事象の後に起こりうる事象を発見することが目的である。このような検索を繰り返し、関連性の高いノード候補を発見することで、因果モデルの作成支援に利用することができる。

4. おわりに

本稿では、ツイートデータを元に、因果モデルの作成を支援し、また、因果モデルに従ったツイートパターンを監視する Agent を制御するシステムについて述べた。本システムは、収集したツイートをより詳細な条件で検索でき、因果モデルのノードの抽出を支援する。また、因果モデルのパターンを Agent に監視させることで、因果モデルに従ったパターンが検知された場合には、モデルに従った次の事象を予報することも可能である。

参考文献

- [1] Takeshi Sakaki, Makoto Okazaki and Yutaka Matsuo, "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors", Proceedings of the 19th International Conference on World Wide Web, pp. 851-860 (2010).
- [2] 荒牧 英治, 増川 佐知子, 森田 瑞樹 "Twitter Catches the Flu: 実定性判定を用いたインフルエンザ流行予測", 情報処理学会研究報告. 自然言語処理研究会報告, Vol.201, No.1 pp. 1-8 (2011).