

マイクロブログからの鉄道の運行トラブル発生検出 および継続状況の抽出

Detecting Occurrences and Continuation Statuses of Train Troubles from Microblogs

土屋 圭^{1*} 豊田 正史^{2†} 喜連川 優^{2,3‡}
Kei Tsuchiya¹ Masashi Toyoda² Masaru Kitsuregawa^{2,3}

¹ 東京大学大学院情報理工学系研究科

¹ Graduate School of Information Science and Technology, The University of Tokyo

² 東京大学生産技術研究所

² Institute of Industrial Science, The University of Tokyo

³ 国立情報学研究所

³ National Institute of Informatics

Abstract: When train troubles occur, users of microblogs often post information about these troubles immediately. Microblogs often include more fine-grained information of train troubles than official information. Such information could make our decision better. In this paper, we try to extract two types information about train troubles by analyzing Twitter. First is detecting train troubles' occurrences. In cases of whole line suspensions and troubles including whole line suspensions and partial suspensions, about 70 percent of train troubles were detected within 500 seconds by using real-time burst detection method. Second is estimating times when train troubles end. To predict remaining time of train troubles, we build some regression models. We find that regression models which take account of past information work better.

1 はじめに

近年、スマートフォンやソーシャルネットワークサービスの普及によって、リアルタイム性のある、詳細な情報がウェブにアップロードされるようになった。スマートフォンの普及率は世界中で増加の一途を辿っており、特に日本においては2016年までに70%を超えると推定されている。世界中で広く利用されているソーシャルネットワークサービスの1つであるTwitterでは、2012年6月に1日の投稿件数が4億を超えた。また、Twitterのアクティブユーザの60%がモバイル端末から利用しているということも分かっている。人々はTwitterのようなリアルタイムに情報を共有するウェブサービスを、スマートフォンによっていつでも、ど

こでも利用することができる。その結果、実世界で発生した出来事は即座にウェブ上のコンテンツに反映される。

我々は実世界の出来事の情報テレビや新聞、ニュースサイト、ソーシャルメディアなど様々なメディアを通じて知ることができる。特にソーシャルメディアには、他のメディアよりも実世界の動きが即座に反映されることや、他メディアの報道内容を補完するような、より詳細な情報を含む場合があるという特徴がある。例えば、地震や台風などの自然災害、交通渋滞、鉄道の運行トラブルなどが発生すると、ソーシャルメディアには現場の状況が即座に反映される。実世界の出来事について、リアルタイムに詳細に把握することは意思決定を下す際に非常に重要である。そこで、ソーシャルメディアから実世界の出来事の情報抽出するという機運が高まっている。

本稿では、公式情報よりも詳細な情報がリアルタイムにソーシャルメディアに反映されている出来事として鉄道の運行トラブルに焦点を当てる。Twitterの投稿を解析することによって、東京メトロ9路線の運行ト

*連絡先: 東京大学大学院情報理工学系研究科電子情報学専攻
〒113-8656 東京都文京区本郷7-3-1
E-mail: tsuchiya@tkl.iis.u-tokyo.ac.jp

†連絡先: 東京大学生産技術研究所
〒153-0041 東京都目黒区駒場4-6-1
E-mail: toyoda@tkl.iis.u-tokyo.ac.jp

‡連絡先: 国立情報学研究所
〒101-8430 東京都千代田区一ツ橋2-1-2
E-mail: kitsure@tkl.iis.u-tokyo.ac.jp

ラブルを対象に主に2つの評価実験を行う。1つ目はバースト検出手法を用いた運行トラブルの発生検出である。運行トラブルの発生をいち早く知ることができれば、そのトラブルから受ける影響を最小限にとどめることができる。2つ目は、運行トラブルが終了するまでの時間の推定である。発生しているトラブルがどの程度継続するかを知ることができれば、今後の予定を中止にしたり、他の交通機関を利用するなどの意思決定が可能となる。

本稿の構成は以下の通りである。2章では鉄道の運行トラブルに関連するツイートを抽出する方法について述べる。3章では既存のバースト検出手法を用いた運行トラブル発生検出を行う。4章では運行トラブル継続状況を推定する手法について提案し、評価実験を行う。5章ではソーシャルメディアから実世界の出来事の抽出を行った関連研究について述べ、最後に6章で本稿をまとめる。

2 鉄道運行トラブル関連ツイートの抽出

本章ではTwitter APIを用いて収集した東京メトロの路線名を含むツイートから、運行トラブルに関連するツイートを抽出する方法について述べる。なお、ツイートの正規化のため、収集した全てのツイートに対して、i) 検索時に使用した路線名を“QUERYWORD”という文字に置換、ii) 投稿に含まれるURLを“URL”という文字に置換、iii) 投稿に含まれる@から始まるユーザ名を“MENTION”という文字に置換、という処理を行った。

2.1 分類器の作成

路線名を含むツイートには、“今、千代田線が全線で運転見合わせている”というように、現在の運行トラブルについて述べているものが含まれている。しかし、“これから千代田線に乗る”のように運行トラブルとは関係ないツイートや、“今夜は千代田線遅れるだろう”のように現在の運行トラブルについて述べていないツイートも存在するため、ツイートが現在の運行トラブルについて述べているかどうかを分類する必要がある。そこで、SVMを用いてツイートの分類を行った。SVMのライブラリにはLIBSVM¹、SVMのカーネルには線形カーネル、特徴量にはツイートに出現する形態素を用いた。また、日本語の形態素解析器にはMeCab²を用いた。SVMの学習データを作成するために、5000件

のツイートに対して人手でラベル付けを行った。これらのツイートは2011年9月21日³、2012年4月3日⁴、2012年9月30日⁵の3日間に投稿された、東京メトロのいずれかの路線名を含むツイートの中からランダムにサンプリングしたものである。ラベルは、 T_0 : 運行トラブル状況について言及なし、 T_1 : 全線運転見合わせ、 T_2 : 一部区間運転見合わせ、 T_3 : 遅延や直通運転中止など運転見合わせ以外の異常の4種類である。ラベル付けの結果を表1に示す。

	T_0	T_1	T_2	T_3
#tweet	3,010	868	461	661

表1: ラベル付けの結果

2.2 分類器の性能評価

次に分類器の性能評価のためにラベル付けを行ったツイートに対し、2値分類の10分割交差検定を行った。表2に結果を示す。単一のトラブルを対象とした場合は T_1 はF値が7割弱であるのに対し、 T_2 、 T_3 は6割前後と T_1 よりも性能が下がる結果となった。また、複数ラベルを同一のトラブルと見なし、トラブルの粒度を粗くすると、分類器の性能が上がることも確認できた。SVMの特徴量の重みや誤分類についての考察は[1]を参照されたい。

Target Label	Precision	Recall	F-value
T_1	0.6931	0.6740	0.6834
T_2	0.6117	0.5859	0.6000
T_3	0.5836	0.5386	0.5602
$T_1 + T_2$	0.7236	0.7111	0.7173
$T_1 + T_2 + T_3$	0.7671	0.7628	0.7649

表2: 10分割交差検定の結果

3 鉄道運行トラブルの発生検出

本章では、鉄道運行トラブルの中でも、 T_1 : 全線見合わせ、 T_2 : 一部見合わせ、 $T_1 + T_2$: 見合わせの3種類のトラブルを対象に、バースト検出手法を用いた運行トラブルの開始時間を抽出する方法について述べ、評価実験を行う。対象とするトラブルは2011年3月から2013年10月までに発生した、開始時間の正解データを分単位で取得できたトラブルのみとした。表3に対象としたトラブルの数を示す。

³台風15号の影響で全路線で見合わせや遅延などが生じた。

⁴強風の影響で銀座線を除く全線で遅延などが生じた。

⁵台風17号の影響で6路線で遅延などが生じた。

¹<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

²<http://mecab.sourceforge.net/>

	T_1	T_2	$T_1 + T_2$
#case	52	39	88

表 3: 運行トラブルのケース数

3.1 バースト検出手法

バースト検出手法には蝦名らの手法 [2] を用いた。蝦名らの手法は Zhang らの手法 [3] で提案された Aggregation Pyramid を参考としている。Zhang らの手法は一定期間ごとにバースト解析を行っているのに対し、蝦名らの手法はイベントが発生するごとにバースト解析を行ため、イベントが発生していない期間の無駄な計算を削減できる。そのため、リアルタイムなバースト検出が可能になり、鉄道運行トラブルという、リアルタイム性が重要となるイベントの発生検出に適していると言える。実験では、2.1 節で作成した分類器によって Positive と判定されたツイートの出現をイベントと定義する。また、この手法によってバーストと判定されたセルの到着時間で最も早いものを、トラブル開始時間の推定値とする。

3.2 パラメータの調整

蝦名らの手法では、予め 4 種類のパラメータを与える必要がある。パラメータはそれぞれ、Aggregation Pyramid の高さ N 、バースト検出の判定基準 β 、Aggregation Pyramid の 1 つのセルにイベントを圧縮する間隔 W_{min} 、バーストと判定するイベント発生間隔の最低個数 A_{min} を決定する。本実験では W_{min} は 1 に固定した。残りのパラメータの調整を行うために、あるトラブル n に対してパラメータの組 θ を与えたときの、バースト検出の結果の評価値を式 (1) によって定義する。

$$score_n(\theta) = |T_{actual} - T_n(\theta)| \quad (1)$$

ただし、 T_{actual} は実際の運行トラブルの開始時間、 $T_n(\theta)$ はパラメータ θ が与えられたときにトラブル n においてバーストと判定されたセルの到着時間のうち最も早い時間である。また、時間は全て秒単位である。あるトラブルのケース n と同種類のトラブルが全部で K ケースあるとき、ケース n のバースト検出には式 (2) を最小化するようなパラメータを用いる。

$$\sum_{i=1, i \neq n}^K score_i(\theta) \quad (2)$$

パラメータの探索は N と A_{min} をそれぞれ 1 から 10 まで 1 刻み、 β を 0.01 から 1.00 まで 0.01 刻みで変化させるグリッドサーチによって行った。

3.3 バースト検出の評価実験

蝦名らの手法はバースト検出を行う前に Aggregation Pyramid のデータ構造内に一定のデータを蓄積する必要がある。バースト検出対象の日に投稿されたツイートを用いて Aggregation Pyramid を構築すると、その日の早い時間帯で実際のバーストがあった際にバースト検出ができなくなる。そこで、 $\frac{20 \times 3600}{AVG(m) \times FPR}$ 秒間隔で到着するツイートで予め Aggregation Pyramid を構築した状態からバースト検出を行った。ここで、 $AVG(m)$ は 2012 年に投稿された東京メトロの路線名 m を含むツイート数の平均値、 FPR は分類器の False Positive Rate である。すなわち、東京メトロが営業行わない午前 1 時から午前 5 時までの約 4 時間を除いた 20 時間における、Positive ツイートの平均到着間隔で予め Aggregation Pyramid を構築する。

3.2 節の方法によって各運行トラブルの各ケースに対してパラメータを求め、バースト検出を行った。比較手法には、Positive ツイートの 5 分間隔のヒストグラムに対して閾値を設定し、その閾値を超えた区間の終了時間をバーストとする手法を用いた。閾値は、蝦名らの手法のパラメータの求め方と同様にして求めた。式 (1) の平均値を表 4 に示す。全ての場合において、バースト検出手法を用いた方が平均評価値が良い結果となった。特に $T_1 + T_2$ においては、バースト検出手法の平均評価値が閾値による手法と比べて半分以下になった。

	T_1	T_2	$T_1 + T_2$
Burst Detection	3753.24	2797.64	2807.05
Threshold	4305.88	3767.69	5682.95

表 4: バースト検出の平均評価値

次に、各ケースの評価値の分布を図 1, 2 に示す。バースト検出手法による T_1 , $T_1 + T_2$ の検出では、500 秒以内にトラブルの発生を検出できたケースが約 7 割を占めている。一方で、これらの平均評価値が数千秒となった原因は、検出した時間が正解時間から大幅にずれたケースの影響を受けているためである。2011 年 9 月 21 日の東西線の全線見合わせもこの例の 1 つである。このケースの Positive ツイートのヒストグラムと共にバースト時間をプロットしたものを図 3 に示す。この例では正解バースト時間よりも早期に発生した、小さなバーストを検出したことによって評価値が 40223 と非常に高くなっている。この例のように、実際のバースト時間よりも早い段階で発生したバーストを検出することによって評価値が大幅に大きくなるケースが多く見られた。このようなケースは、大規模な運行トラブルが発生し、その日に投稿される Positive ツイートの数が他のケースと比べて急激に増加した場合に多く発生する傾向があった。先の例の 2011 年 9 月 21 日は

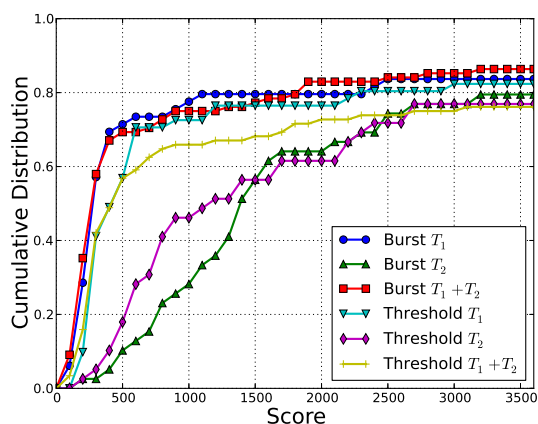


図 1: 評価値の 1 時間以内の分布

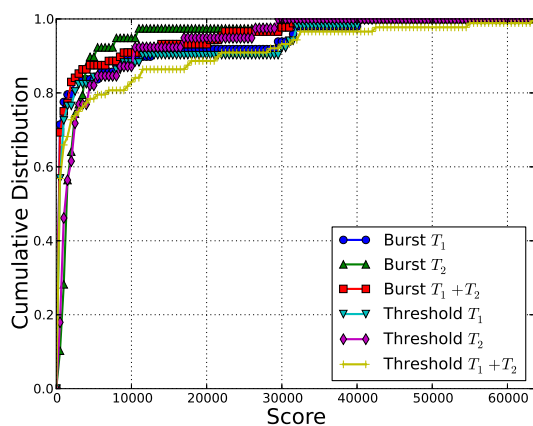


図 2: 評価値の全体の分布

台風 15 号の影響で、東西線を含む首都圏の交通機関が大きく乱れた。

4 鉄道運行トラブルの継続状況抽出

本章では、鉄道の運行トラブルが発生した場合に、その運行トラブルが終了するまでに見込まれる時間を推定する手法について述べる。

4.1 回帰モデル

提案手法では、運行トラブルが終了するまでの分単位の残り時間を従属変数とした回帰モデルによって残り時間の推定を行う。運行トラブルが発生してから 5 分ごとに、その時間帯に投稿されたツイートを分類し特徴量を算出する。本稿では、特徴量の算出方法が異なる 4 つの手法で実験を行った。以下、 \mathbf{x}_i は i 番目の

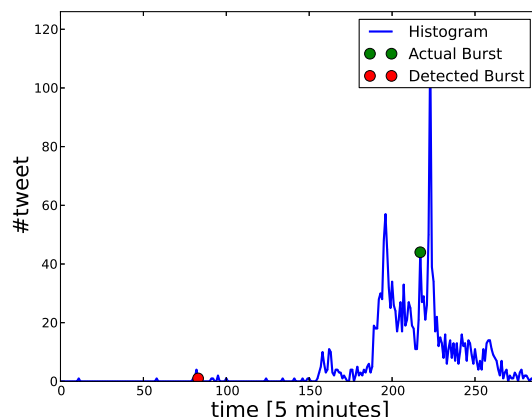


図 3: 2011 年 9 月 21 日の東西線の全線見合わせ発生時のヒストグラムおよびバースト時間

時間帯に投稿された全 Positive ツイートの形態素の出現頻度を表すベクトルとする。

1 つ目の手法 (以下, Single) は、直近の時間帯のみの情報を用いる手法で

$$\mathbf{x}_n \quad (3)$$

によって n 番目の時間帯の特徴量を求める。

2 つ目の手法 (以下, Addition) は運行トラブル発生以降、各時間帯の特徴量を順次足していく手法である。 n 番目の時間帯の特徴量は

$$\sum_{i=1}^n \mathbf{x}_i \quad (4)$$

によって計算される。

3 つ目の手法 (以下, Window) はウィンドウサイズ W を設定し、 W だけ過去の時間帯の特徴量を加える手法である。 n 番目の特徴量は

$$\begin{cases} \frac{W}{n} \sum_{i=1}^n \mathbf{x}_i & (1 \leq n \leq W) \\ \sum_{i=n-W+1}^n \mathbf{x}_i & (n > W) \end{cases} \quad (5)$$

によって求められる。

4 つ目の手法 (以下, Exponent) は過去の時間帯の特徴量も織り交ぜながら、直近の時間帯の特徴量が重視されるように特徴量を計算する手法である。 n 番目の時間帯の特徴量は

$$\sum_{i=1}^n \mathbf{x}_i e^{-\theta(n-i)} \quad (6)$$

によって算出される。ここで、 θ は過去の時間帯の特徴量をどの程度織り込むかを調整するパラメータである。

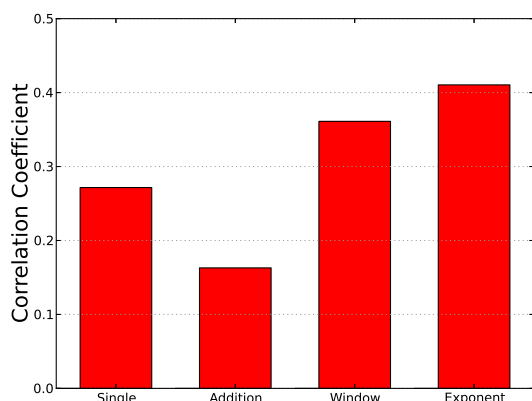


図 4: 実際の残り時間と推定した残り時間の相関係数

4.2 回帰モデルの相関係数

$T_1 + T_2$: 運転見合わせを対象に、提案手法の評価実験を行った。回帰モデルの学習は、推定を行うケース以外の全てのケースで行った。回帰モデルには L2-regularized L2-loss SVR, ライブラリには LIBLINEAR⁶を用いた。Window のウィンドウサイズは 3 とした。また、Exponent のパラメータ θ は 0.0 から 1.0 まで 0.1 刻みで実験を行い、相関係数が最も高くなった $\theta = 0.2$ を用いた。

実際の残り時間と推定した残り時間の相関係数を図 4 に示す。過去の時間帯の特徴量を、直近の時間帯の特徴量と同等に扱う Addition が最も相関が低くなった。一方で、過去の情報のある程度限定して用いる Window, Exponent は、過去の情報を一切使用しない Single に比べ、相関係数が高くなった。このことから、直近の時間帯の情報に加えて、過去の情報を考慮することの有効性が確認できた。

4.3 継続時間推定の評価実験

4.1 節で提案した回帰モデルによって得られたトラブル収束までの残り時間が t 分以上かどうかで、そのトラブルが以後 t 分以上継続するかどうかを判定する実験を行った。なお、Window のウィンドウサイズおよび Exponent のパラメータ θ は相関係数の実験と同様に、順に 3, 0.2 とした。本実験では、 t は 15 分, 30 分, 45 分, 60 分の 4 通りを試みた。また、ベースラインには実際の残り時間が t 分以上であるケースが半数以上の場合には全て t 分以上、そうでない場合には全て t 分未満と判定する手法を用いた。

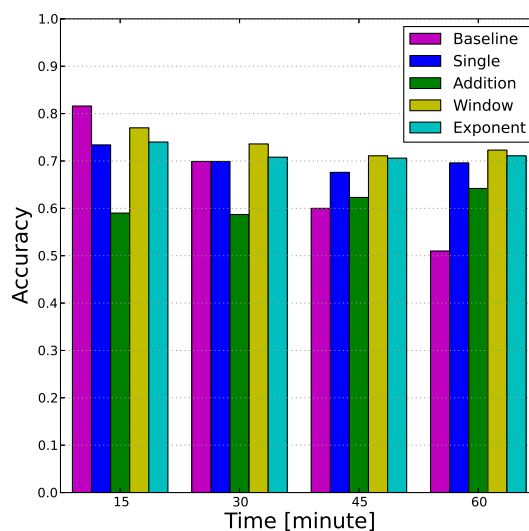


図 5: 運行トラブル継続時間判定の精度

図 5 に実験で得られた運行トラブル継続時間判定の精度を示す。継続時間が 45 分以上のケースを判定する場合、提案した全ての回帰モデルにおいて、ベースラインよりも精度が高くなった。特に Window と Exponent は全ての時間において精度が 7 割以上であり、提案手法の有効性が確認できた。相関係数は Window よりも Exponent の方が高い結果であったが、継続時間の判定においては、全ての場合において Window の精度が高い結果となった。また、Window および Exponent の精度は、全ての場合において Single の精度よりも高くなり、過去の情報のある程度限定して織り込むことの有効性が確認できた。

5 関連研究

ソーシャルメディアを解析することによって、実世界の出来事を抽出するという研究は多岐に渡っている。人の行動パターンや状態などを対象にした研究として、Eytan ら [4] によるインターネットユーザの行動を予測する研究や、Adam ら [5] による個々人の健康状態を予測する研究がある。Eytan らは、過去のユーザの検索履歴を解析して、インターネットユーザの今後の行動を予測した。Adam らは Twitter の投稿内容、投稿位置、ソーシャルリンクを解析して、Twitter のユーザの健康状態を最大 7 日先まで予測した。次に、社会や経済の動きを抽出する研究として、Eric ら [6] によるソーシャルメディアと株価変動の相関を調査した研究や、Kira ら [7] による今後のニュースを予測する研究、Panagiotis ら [8] による選挙結果を予測する研究があ

⁶<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

る。Eric らはブログ上の不安表現と株価指数の相関を見つけ、株価変動の予測を行った。Kira らは過去 150 年分のニュース記事から因果関係を抽出し、将来の出来事を予測した。Panagiotis らは Twitter を解析して、選挙結果の予測を行った。自然現象を対象とした研究には、Sakaki ら [9] の地震と台風を対象にした研究や、Hapipeng ら [10] の積雪や植生を対象にした研究がある。Sakaki らは Twitter の投稿内容と投稿位置を用いて、地震と台風それぞれの発生時間および発生位置を推定した。Haipeng らは写真共有サイトである Flickr に投稿された写真およびそのタグを解析し、積雪の有無や植生を抽出する手法を提案した。

本稿で対象とした鉄道運行トラブルは、全線見合わせ、一部見合わせ、遅延、直通運転中止などイベントに複数の種類があることや、トラブルの連鎖が起こりうること、同時多発的に発生する可能性があることなどから先行研究が対象としているイベントに比べて複雑であると言える。このような複雑な問題に対する情報抽出は、この分野の課題であると言える。

6 おわりに

ソーシャルメディアから有用な情報を抽出する研究への期待が高まる中、本稿では鉄道の運行トラブルに焦点をあて、Twitter を解析することによって主に 2 種類の情報抽出を行った。運行トラブルの発生検出では、蝦名らが提案したリアルタイムバースト検出手法を用いることで、運行トラブルの開始時間の抽出を試みた。特に T_1 , $T_1 + T_2$ を対象とした場合は、それぞれ約 7 割のトラブルの発生を 500 秒以内に検出することができた。運行トラブルの継続状況の抽出については、回帰モデルを用いてトラブル収束までの残り時間を推定する手法を提案した。実験の結果から、直近のツイートのみでなく、過去の情報も織り交ぜることの有効性が確認できた。今後は本稿では扱わなかった鉄道の運行トラブルに関する情報を抽出したいと考えている。本稿では路線名を含むツイートのみを用いたが、例えば駅名を含むツイートも解析対象に加えることで、運行トラブルの発生場所を抽出できる可能性がある。本稿で提案した手法の改良に加えて、様々な可能性を模索していきたい。

参考文献

- [1] 土屋 圭, 豊田 正史, 喜連川 優: マイクロブログを用いた鉄道の運行トラブル状況抽出に関する一検討, *IFAT*, Vol. 111, No. 31 (2013)
- [2] 蝦名 亮平, 中村 健二, 小柳 滋: リアルタイムバースト検出手法の提案, *日本データベース学会論文誌*, Vol. 9, No. 2, (2010)
- [3] Zhang, X., Shasham D: Better Burst Detection, *ICDE'06 Proceedings of the 22nd International Conference on Data Mining*, pp. 146–149 2006,
- [4] Eytan Adar, Daniel S. Weld, Brian N. Bershad and Steven D. Gribble: Why We Search: Visualizing and Predicting User Behavior, *WWW*, 2007
- [5] Adam Sadilek, Henry Kautz and Vincent Silenzio: Predicting Disease Transmission from Geo-Tagged Micro-Blog Data, *AAAI*, 2012
- [6] Eric Gilbert and Karrie Karahalios: Widespread Worry and the Stock Market, *AAAI*, 2010
- [7] Kira Radinsky, Sagie Davidovich and Shaul Markovitch: Learning Causality for News Events Prediction, *WWW*, 2012
- [8] Panagiotis T. Metaxas and Eni Mustafaraj: How (Not) to Predict Elections, *IEEE third international conference on social computing*, 2011
- [9] Takeshi Sakaki, Makoto Okazaki and Yutaka Matsuo: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors, *WWW*, 2010
- [10] Haipeng Zhang, Mohammed Korayem and David J. Crandall: Mining Photo-sharing Websites to Study Ecological Phenomena, *WWW*, 2012