A Probabilistic Approach for Events Identification from Social Media RSS Feeds

Chiraz Trabelsi[1] and Sadok B. Yahia[1,2]

[1] Faculty of Sciences of Tunis, University Tunis El-Manar, 2092 Tunis, Tunisia
[2] Department of Computer Science, TELECOM SudParis, UMR 5157 CNRS Samovar, 91011
Evry Cedex, France
{chiraz.trabelsi,sadok.benyahia}@fst.rnu.tn

Abstract. Social Media RSS feeds are the most up-to-date and inclusive releases of information on current events used by the new social media sites such as Twitter and Flickr. Indeed, RSS feeds are considered as a powerful realtime means for real-world events sharing within the social Web. By identifying these events and their associated social media resources, we can greatly improve event browsing and searching. However, a thriving challenge of events identification from such releases is owed to an efficient as well as a timely identification of events. In this paper, we are mainly dealing with event identification from heterogenous social media RSS feeds. In this respect, we introduce a new approach in order to get out these events. The main thrust of the introduced approach stands in achieving a better tradeoff between event identification accuracy and swiftness. Specifically, we adopted the probabilistic Naive Bayes model within the exploitation of stemming and feature selection techniques. Carried out experiments over two real-world datasets emphasize the relevance of our proposal and open many issues.

Keywords: Event Identification, Social Media, RSS feeds, Naive Bayes Model.

1 Context and Motivations

Social media sites such as Twitter and Flickr are using RSS[1] feeds for sharing a wide variety of current and future real-world events. Indeed, RSS feeds are considered as a powerful means of communication for social websites looking to share information on a wide variety of real-world events. These events range from popular, widely known ones, *e.g.,* a concert by a popular music band, to smaller scale local events, *e.g.,* a local social gathering, a protest, or an accident. Social media RSS feeds can typically reflect these events as they happen. By identifying all these events, and their associated social media resources flagged in the RSS feeds items, *e.g.,* photographs, videos, etc., we can enable powerful local event browsing and search, to complement and improve the local search tools that Web search engines provide. In this paper, we address the problem of how to identify, accurately et efficiently, events and their related social media resources over social media RSS feeds items. Actually, social media RSS feeds are defined as a collection of informal data that arrives over time and each RSS feed item is associated

[1] Really Simple Syndication.

B. Hong et al. (Eds.): DASFAA Workshops 2013, LNCS 7827, pp. 139–152, 2013.

with some social attributes (features) such as title, description and tags. The content of such RSS feed item, generated from social websites, is particularly useful for real-time identification of real-world events and their associated social media resources, which is the problem that we address in this paper.

Overall, social RSS feeds items, generally exhibit information that is useful for identifying the related events, if any, but this information is far from uniform in quality and might often be fragmented and noisy. Our problem is most similar to the event detection task on Twitter social media [4, 5, 16, 1, 15], whose objective is to identify events in a continuous stream of news documents *e.g.,* earthquakes (Sakaki *et al.,* 2010) or news events (C. Aggarwal and K. Subbian, 2012). However, our problem exhibits some fundamental differences from these event detection approaches that originate in the social RSS feeds resources on which we focus. Specifically, event identification aims to discover and cluster events found in textual news articles. These news articles adhere to certain grammatical, syntactical, and stylistic standards that are appropriate for their venue of publication. Therefore, most state-of-the-art event detection approaches leverage natural language processing tools such as named-entity extraction and part-of-speech tagging to enhance the document representation [4, 5, 1]. In contrast, social RSS feeds resources contain little textual narrative, usually in the form of a short description, title, or keyword tags. Importantly, the lack of both text content and implicite network structure within social RSS feeds resources such as for Twitter messages, renders traditional event identification techniques unsuitable over social RSS feeds resources.

In this paper, we introduce a new approach, called RssE-Miner, for online real-world event identification, that efficiently exploit the Naive Bayes Model for identifying events and their related social RSS feeds resources. The proposed approach consists of three steps: 1) Data preparation and feature selection; 2) Model learning and; 3) Event identification. The main originality of RssE-Miner stands in achieving a meaningful tradeoff between runtime performance and event identification accuracy from social media RSS feeds.

The remainder of the paper is organized as follows. Section 2 thoroughly scrutinizes the related work. We describe later in Section 3 our probabilistic approach for events identification from social media RSS feeds composed of three major steps. An illustrative example is also provides for underpinning the different steps of our approach. The experimental study of our approach is illustrated in Section 4. Section 5 concludes this paper and sketches avenues for future work.

2 Related Work

We describe related work in three areas namely event identification or detection, RSS Feeds studies and Naive Bayes applications.

The event detection task [14, 12] was studied on a continuous stream of news documents with the aim to identify news events and organize them. This is one of the important tasks considered by the topic detection and tracking (TDT) [2]. The topic detection and tracking (TDT) event detection task [14] was studied in a notable collective effort to discover and organize news events in a continuous stream, e.g., newswire, radio broadcast, etc. [16, 1] With an abundance of well-formed text, many of the proposed approaches, *e.g.,* [5] rely on natural language processing techniques to extract linguistically motivated features. Hogenboom et *al.,* [9] proposed an approach for detecting economic events using a semantics-based pipeline. They noted that augmenting documents with semantic terms did not improve performance, and reasoned that inadequate clustering techniques were partially to blame. In our setting, we improve the event identification performance with a judiciously combination of the social media features. Becker et al., in [4], proposed an approach for event detection that aims to partition a set of collected documents from social media into clusters such that each cluster corresponds to all documents that are associated with one event, which is similar to our approach. However, They start from a re-interpretation of exemplar-based SVMs, while we focus more on a probabilistic perspective and Naive Bayes Model. They also propose an additional step for the event detection task during which they use a document similarity metrics to enable online clustering of media to events.

Several efforts have focused on RSS Feeds processing: We can talk about SPEED (Semantics-Based Pipeline for Economic Event Detection) [10] which is a framework that aims at extracting financial events from news articles (announced through RSS Feeds) and annotated with meta-data that makes real-time use possible. It's modeled as a pipeline that reused some of the ANNIE GATE components and develop new ones. We can also refer to SemNews system [11] which is a Semantic Web-based application that aims at accurately extracting information from heterogenous sources. It seeks to discover the meaning of news items. These items are retrieved from RSS Feeds and are processed by the NLP engine OntoSem. Our work also involves the processing of RSS Feeds but using the statistical approach rather than the linguistic approach. The latter constitute a hindrance since linguistic expert uses rules for manually defining event patterns, which is prone to errors.

The literature witness a various applications of the Naive Bayes model. Actually, Naive Bayes showed a powerful performances for automatic categorization of email into folders [6], where email arrives in a stream over time. It was mentioned that Naive Bayes is the fastest algorithm compared to, respectively, MaxEnt, SVM and Winnow. Naive Bayes was also used as a pre-trained model for real-time network traffic classification [13]. Furthermore, Naive Bayes represents, yet, one of the most popular machine learning models applied in the spam filtering domain [17]. Importantly, the learning process of Naive Bayes is extremely fast compared with current discriminative learners, which makes it practical for large real-world applications. Since the training time complexity of Naive Bayes is linear to the number of training data, and the space complexity is also linear in the number of features, it makes Naive Bayes both time and storage efficient for practical systems. This led us to opt for the choice of the Naive Bayes algorithm for social media RSS Feeds processing.

3 Events Identification from Social Media RSS Feeds

As we previously mentioned, identifying social events from social RSS feeds items is a compelling issue for the efficient use of the social RSS feeds releases. Indeed, such events can be useful not only for humans, but also to software agents and applications on the social web. Our hypothesis is that this underlying knowledge can be derived by means of Naive Bayes classifier combined with adequate algorithms for stemming and feature selection. Therefore, the most salient features of our approach, namely RssE-Miner, are as follows: *(i)* It is a domain independent approach, since no domain assumptions are formulated and no predefined knowledge is needed and; *(ii)* It relies on the Naive Bayes model for identifying the RSS feed items related to the same event.

Problem definition: Given a set of social RSS feed resources associated with events, the problem that we address in this paper is how to identify the events that are reflected in the social RSS feed resources, and to correctly assign the resources that correspond to each event. We cast our problem as a classification problem over social RSS feed resources, *e.g.,* photographs, where each social RSS feed resource includes a variety of "features" with information about the resource. Some of these features, *e.g.,* title, description, tags, are manually provided by users, while other features, *e.g.,* upload or con- tent creation time, are automatically generated by the social website. As the formal definition of "event", we adopt the version used for the Topic Detection and Tracking (TDT) event detection task over broadcast news [18].

Definition 1. (EVENT) *An event is something that occurs in a certain place at a certain time.*

In what follows, we introduce the whole process of the proposed RssE-Miner approach. It consists of three steps: 1) Data preparation and feature selection; 2) Model learning and; 3) Event identification. At a glance, Figure 1 provides a visual representation of the RssE-Miner approach.

3.1 Data Preparation and Feature Selection

While social RSS feeds resources present challenges for event detection, they also exhibit opportunities not found in traditional news articles. Specifically, social RSS feeds resources usually have a wealth of associated features such as, *e.g.,* title, description, tags, as well as automatically generated information, *e.g.,* content creation time. Individual feature might be noisy or unreliable, but collectively they provide revealing information about each social RSS feed resource, and this information is valuable to address our problem of focus. There are many design choices for feature representation. In this paper, we made use of the conventional bag of words representation of text items. Hence, each RSS feed item, is represented as a bag of words $\{w_1, w_2, \ldots, w_k\}$. In this respect, common refinements techniques such as discarding common stop-words, pronouns, articles, prepositions and conjunctions are applied. A stemming algorithm provided by Lucene[2] is also used for reducing morphologically similar words, *e.g.,* "starting", "starts" and "start", to the root word, *i.e.,* "start".

Thereafter, we proceed to the feature selection stage in order to identify the most salient features for our model learning. For this purpose, we made use the CFS (Correlation based Feature Selection) algorithm proposed by Hall et *al.,* in [8, 7]. Actually,

[2] An open-source search engine that provides full text indexing and searching capabilities, `http://lucene.apache.org/`

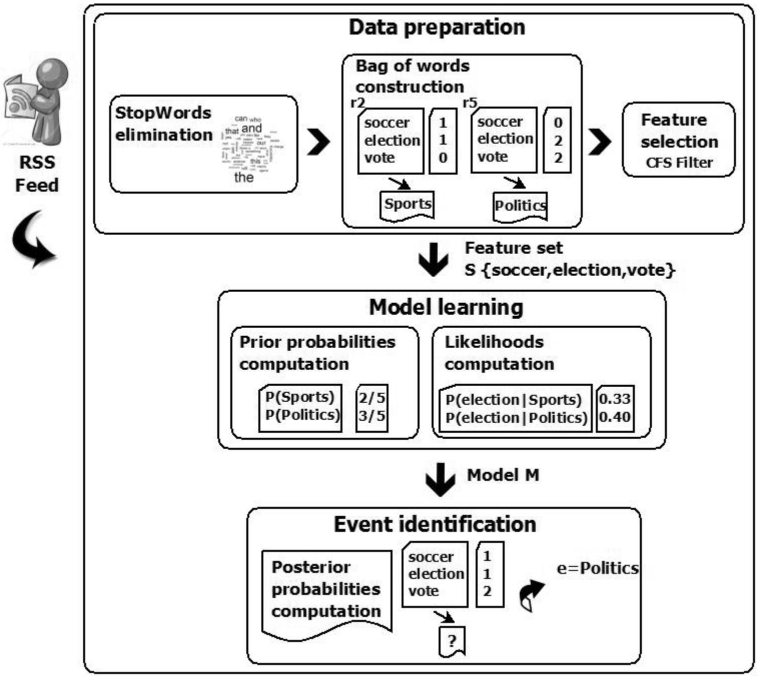

Fig. 1. The RssE-Miner approach at a glance

CFS is a simple filter algorithm[3] that ranks feature subsets according to a correlation based heuristic evaluation function. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other. Irrelevant features are ignored because they will have low correlation with the class. While, redundant features are screened out as they will be highly correlated with one or more of the remaining features. CFS algorithm is based on a heuristic evaluation function defined as follows :

$$M_S = \frac{k\overline{h_{cf}}}{\sqrt{k + k(k-1)\overline{h_{ff}}}}. \tag{1}$$

where M_S is the heuristic "merit" of a feature subset S containing k features, $\overline{h_{cf}}$ is the mean feature-class correlation ($f \in S$), and $\overline{h_{ff}}$ is the average feature-feature intercorrelation. The numerator of Equation (1) can be thought of as providing an indication of how predictive of the class a set of features are; the denominator of how much redundancy there is among the features. In this paper, Best first heuristic search is used [8].

[3] The time complexity of CFS is quite low. It requires $m((n^2 - n)/2)$ operations for computing the pairwise feature correlation matrix, where m is the number of instances and n is the initial number of features.

3.2 Model Learning

The second step of RssE-Miner aims to determine the likelihood of a RSS feed resource r_i for which the target event is e ($e \in$ a set of events $\mathcal{E}$), given the bag of word $\mathcal{W}$ representing r_i. The variable $\mathcal{W}$ represents a bag of words $\{w_1, w_2, \ldots, w_k\}$ that are extracted from a RSS feed r_i. Hence, a RSS feed resource (represented by $\mathcal{W}$) is identified to be related to an event e_i if the following expression holds:

$$\forall j \neq i \quad \frac{P(\mathcal{W}|e_i)}{P(\mathcal{W}|\bar{e}_i)} > \frac{P(\mathcal{W}|e_j)}{P(\mathcal{W}|\bar{e}_j)} \tag{2}$$

Where $P(\mathcal{W}|e)$ represents the likelihood that $\mathcal{W} = \{w_1, w_2, \ldots, w_K\}$ is produced given the event is e, and $P(\mathcal{W}|\bar{e})$ represents the likelihood that $\mathcal{W}$ is produced given the event is not e. In the Naive Bayes approach, statistical independence is assumed between each of the individual words in $\mathcal{W}$. Under this assumption, the likelihood of $\mathcal{W}$ given an event e is approximated as:

$$P(\mathcal{W}|e) = \prod_{i=1}^{K} P(w_i|e). \tag{3}$$

This expression can be alternatively be represented with a counting interpretation as follows:

$$P(\mathcal{W}|e) = \prod_{i=1}^{K} P(w_i|e)^{C_{w|\mathcal{W}}}. \tag{4}$$

Where $\mathcal{V}$ represents the vocabulary set of words in all training examples. In this interpretation, the occurrence count $C_{w|\mathcal{W}}$ within $\mathcal{W}$ of each word w in the vocabulary set $\mathcal{V}$, is used to exponentially scale the score contribution of that word.

The likelihood function $P(\mathcal{W}|\bar{e})$ is generated as follows:

$$P(\mathcal{W}|\bar{e}) = \frac{1}{K_{\mathcal{E}} - 1} \sum_{\forall e_i \neq e} P(\mathcal{W}|e_i) \tag{5}$$

Where $K_{\mathcal{E}}$ represents the total number of known events. This expression assumes a uniform prior distribution over all events.

In practice the likelihood function $P(w|e)$ is estimated from training examples using maximum *a posteriori* probability (MAP) estimation with Laplace smoothing as follows:

$$P(w|e) = \frac{N_{w|e} + N_V P(w)}{N_{\mathcal{W}|e} + N_V} \tag{6}$$

In this Equation, N_V is the total number of words in the vocabulary used in all training examples, $N_{w|e}$ is the number of times that word w occurs in the training examples related to the event e, and $N_{\mathcal{W}|e}$ is the total number of words in the training examples related to the event e. The term $P(w)$ represents the prior likelihood of word w occurring independently of the event. This likelihood function is also determined using MAP estimation with Laplace smoothing as follows:

$$P(w) = \frac{N_w + 1}{N_{\mathcal{W}} + N_V} \tag{7}$$

In this Equation, N_w is the number of occurrences of the specific word w in the training set examples and $N_{\mathcal{W}}$ is the total count of all words from the N_V word vocabulary in the training set examples.

3.3 Event Identification

Once the model learning step is performed, we proceed with testing the model for identifying the appropriate event. Classification will occurs when event probability is computed. Indeed, the main goal of the event identification step is to find the class label e ($e \in$ a set of events $\mathcal{E}$) which is most likely to generate the RSS feed resource r. Hence, given that the RSS feed resource r is represented as a bag of words $\{w_1, w_2, \ldots, w_k\}$, RssE-Miner tries to assign the most probable class label (event $e \in \mathcal{E}$) to the RSS feed resource based on these words. Therefore, the class label e for r is the most likely event, given the known words of r, is defined as:

$$e = arg \max_{e \in \mathcal{E}} P(e|r) = \arg\max_{e} P(e)P(r|e). \tag{8}$$

Hence, according to Equation 3 and Equation 8, we have:

$$e = \arg\max_{e \in \mathcal{E}} P(e) \prod_{i} P(w_i|e). \tag{9}$$

The event e which have the best score according to Equation 9 is therefore selected.

Example 1. Suppose that once performing the Data preparation and feature selection step, we obtain the model learning example depicted by Table 1. And let us assume that the Laplace smoothing is equal to 1. Then, the prior probabilities for each class label and the probabilities for every word are computed as following:

Table 1. Model learning example

w	soccer	election	vote	Event label
r1	1	0	0	Sports
r2	1	1	0	Sports
r3	0	0	1	Politics
r4	0	1	1	Politics
r5	0	2	2	Politics

Table 2. P($word|Event\ label$) computation

word	Event label	P($word\|Event\ label$)
election	Sports	0.33
election	Politics	0.40
soccer	Sports	0.50
soccer	Politics	0.10
vote	Sports	0.17
vote	Politics	0.50

$$P(Sports) = 2/5.$$

$$P(Politics) = 3/5.$$

The word ”election” occurs 1 times in ”Sports” resources.

The total number of words in ”Sports” resources = 1+1+1= 3. Then, we have:

$$P(election|Sports) = (1 + 1)/(3 + 3) = 1/3.$$

The word ”election” occurs 3 times in ”Politics” resources.

Then

$$P(election|Politics) = (3 + 1)/(7 + 3) = 2/5.$$

Table 2 resumes this stuff.

We proceed by tackling the same example mentioned above for event identification task. Hence, we provide in Table 3 an example of a resource for which we are looking for the associated event.

Table 3. A RSS feed resource example

w	soccer	election	vote	Event label
r6	1	1	2	?

- e_j=Sports :

$$\begin{aligned} P(r6|Sports) &= P(soccer|Sports)P(vote|Sports)^2 P(election|Sports) \\ &= 0.0048. \end{aligned}$$

- e_j=Politics :

$$\begin{aligned} P(r6|Politics) &= P(soccer|Politics)P(vote|Politics)^2 P(election|Politics) \\ &= \mathbf{0.010}. \end{aligned}$$

Then the event with the highest posterior probability, is selected.

$$e_j = Politics.$$

Fig. 1 (page 4) clearly illustrates the different steps of the RssE-Miner approach for treating the aforementioned example.

4 Experimental Results

To evaluate the performances of our approach for events identification from social media RSS feeds, we carried out experiments on two real world datasets collected from the online photo management and sharing application Flickr[4]. In order to analyze the accuracy of our approach we adopted the common evaluation measures, namely *Accuracy*[5], *Precision*[6] and *Recall*[7] [3]. We describe, in what follows, the datasets characteristics and the baseline models used for evaluating our approach. Thereafter, we present the results from our experiments.

[4] http://www.flickr.com/

[5] Ratio of correctly classified instances to the total number of instances in the test set.

[6] The proportion of RSS feeds resources correctly associated to the event e from those associated to e.

[7] The proportion of resources correctly associated to the event e from those actually related to e.

4.1 Datasets Description

We collected our dataset from Flickr using the Flickr API[8]. It consists of RSS Feeds of two real world datasets namely Upcoming and Last.fm.

- **Last.fm (dense dataset) :** It consists of all Flickr photographs that were manually tagged by users with an id corresponding to an event from the Last.fm music event catalog[9]. The Last.fm dataset contains 3356 images spread over 316 unique events. The Last.fm dataset is considered to be dense since it includes only events in the area of music. It is more likely to find many resources per event.
- **Upcoming (sparse dataset) :** It consists of all photographs that were manually tagged by users with an event id corresponding to an event from the Upcoming event database[10]. These Upcoming tags provide the ”ground truth” for our classification experiments. Each photograph corresponds to a single event, and each event is self-contained and independent of other events in the dataset. The Upcoming dataset contains 5778 images spread over 362 unique events. The Upcoming dataset is considered as a sparse dataset since it contains different kind of events which are of public interest. It is less likely to find two or more items that belong to the same event. Indeed, it includes fewer items per event.

4.2 Training Methodology

We train our approach for event identification on data from the Upcoming dataset, and test them on unseen Upcoming data, as well as Last.fm data. We order the photographs in the Upcoming dataset according to their upload time, and then divide them into two equal parts. We use them as training and test sets. Hence, we use the training set to train classifiers for the classification task. The second datasets of the Upcoming data and the Last.fm data are used as test sets, on which we report our results. We chose a time-based split since it best emulates real-world scenarios, where we only have access to past data with which we can train models to classify future data.

4.3 Baseline Models

To the best of our knowledge, domain independent event identification from social RSS feeds have never been modeled before. Thus, for enhancing the effectiveness of our approach, we have selected three baselines models as follows:

- **Most Popular Events Identified:** For each event, we counted in how many resources it occurs and used the resources ranked by event occurrence count. For each event, the resources are randomly selected.
- **Most Popular Event Aware Extracted:** Events are weighted by their co-occurrence with a given event. Then, resources are ordered without validation.

[8] `http://www.flickr.com/services/api/`

[9] `http://www.last.fm/events`

[10] `http://www.upcoming.org`

- **SMO:** We compare our approach vs the Becker et al.(2010) [4] approach (we only make use of its classification-based technique part). In fact, Becker et al.(2010) used SVM as a classifier with similarity scores as features to predict whether a pair of documents belongs to the same event. They selected Weka's sequential minimal optimization implementation. In this respect, We implement Naive Bayes as a part of the Weka[11] software system.

4.4 Efficiency of Our Approach

We report, in the following, the obtained results averaged over 10 test runs. We empirically decide to use only description, title and tags features. Indeed, the presence of other features such as location is an indication of document dissimilarity.

Accuracy: Figure 2 (Left), depicts averages of accuracy on Upcoming dataset. Figure 2 (Right), depicts averages of accuracy on Last.fm dataset. On both datasets, SMO shows the highest accuracies. In fact, SMO outperforms RssE-Miner - by notable 1% in the case of Upcoming dataset and by notable 1% in case of Last.fm dataset, but the difference is not statistically significant. However, the performance of RssE-Miner could likely be improved by applying a more sophisticated smoothing method than Laplace. RssE-Miner accuracy, is acceptable since we are looking for the best tradeoff between runtime performance and classification accuracy.

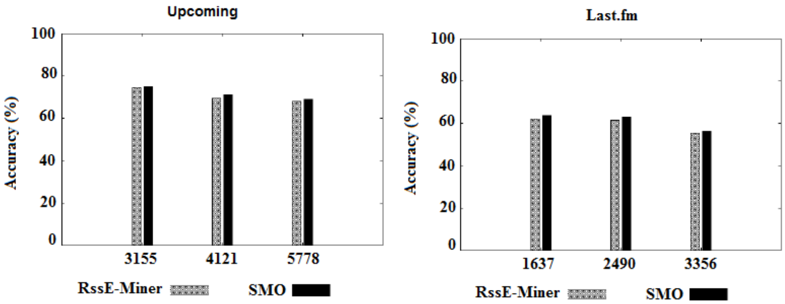

Fig. 2. Left: Averages of *Accuracy* on Upcoming dataset; **Right**: Averages of *Accuracy* on Last.fm dataset

Running Time: As mentioned above, our main goal is to achieve a meaningful tradeoff between runtime performance and classification accuracy. Table 4 depicts that our approach has succeeded in fulfilling this task. Thus, according to these results, we can point out that our approach outperforms SMO approach. In fact, as expected, the Runtime of the SMO approach are much slower than those achieved by our approach for both datasets. We note that RssE-Miner is by far the fastest approach. It takes no more than 2.278(s) on Upcoming dataset and no more than 0.739(s) on Last.fm dataset. In

[11] `http://www.cs.waikato.ac.nz/ml/weka/`

particular, our approach outperforms the SMO approach by a large and statistically significant margin. To this end, RssE-Miner makes real-time use possible. This is of paramount importance in the case of event identification in social media RSS Feeds as faster processing of data enables one to make better informed decisions.

In all, evaluation underlines fast and accurate performance by applying our approach. Indeed, achieved results show that event identification using Naive Bayes model can work in near real-time without obvious decrease in accuracy.

Table 4. Average of runtime on the Upcoming and Last.fm datasets

	Instances	Features	Events	Runtime(s)	
				RssE-Miner	SMO
Upcoming	3155	30	203	**0.596**	34.213
	4121	33	280	**1.215**	66.392
	5778	36	362	**2.278**	115.33
Last.fm	1637	21	171	**0.18**	24.571
	2490	27	243	**0.519**	50.143
	3356	23	316	**0.739**	85.829

4.5 Effectiveness of Our Approach

We present in Fig. 3 and Fig. 4- precision as well as recall measures on Upcoming and Last.fm datasets. Indeed, according to the sketched histograms, we can point out that our approach outperforms both baselines. On Upcoming dataset, the average recall achieves high percentage for higher value of N, *i.e.,* the number of extracted events. Indeed, for N = 58, the average Recall is equal to 0.742, showing a drop of 98.38% compared to the average Recall for N = 36. On Last.fm dataset, the average recall achieves high percentage for higher value of N. Indeed, for N = 46, the average Recall is equal to 0.878, showing a drop of 99.4% compared to the average Recall for N = 41. In this case, for a higher value of N, by matching resources with their corresponding events, the proposed approach can achieve event identification task successfully. In addition, the average precision of RssE-Miner outperforms the two baselines. On Upcoming dataset, our approach achieves the best results when the value of N is around 58. In fact, for N = 58, it has an average of 68.3% showing an exceeding about 4% against the first baseline and around 59.6% against the second one. On Last.fm dataset, our approach achieves the best results when the value of N is around 41. In fact, for N = 41, it has an average of 87.3% showing an exceeding about 12.9% against the first baseline and around 64.5% against the second one. These results highlight that the proposed approach can better improve event identification task even for a high number of extracted events.

4.6 Online Evaluation

We present in Fig. 5 the runtime of RssE-Miner. Since it is hard to measure the exact runtime of the proposed approach, we simulated an online execution of our approach

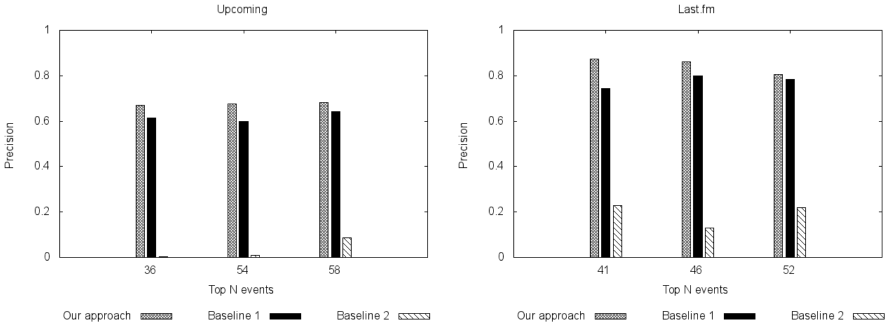

Fig. 3. Left: Averages of *Precision* on Upcoming dataset; **Right**: Averages of *Precision* on Last.fm dataset

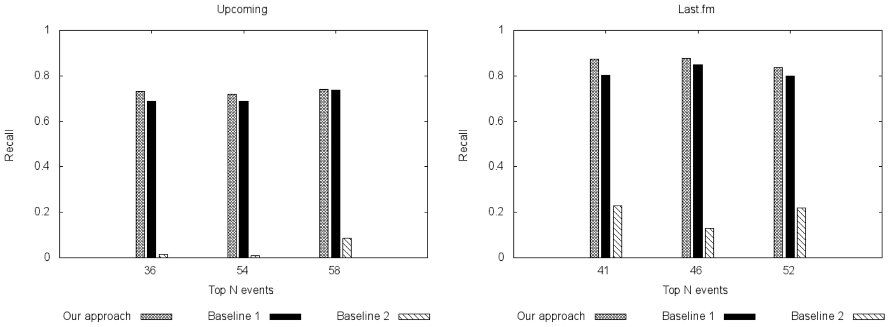

Fig. 4. Left: Averages of *Recall* on Upcoming dataset; **Right**: Averages of *Recall* on Last.fm dataset

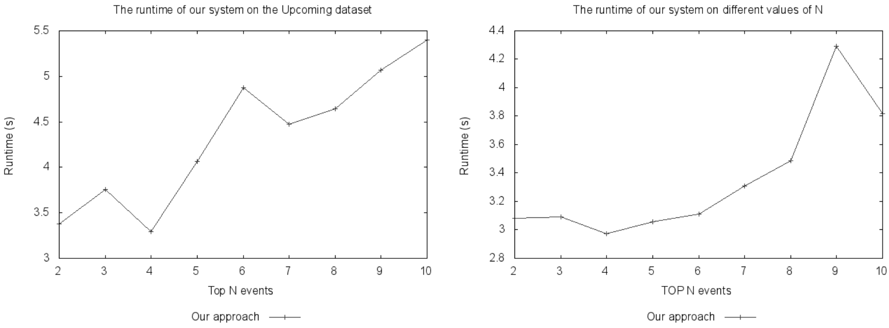

Fig. 5. Left: The runtime of our system online with different values of N on the Upcoming dataset; **Right**: The runtime of our system online with different values of N on the Last.fm dataset

among the Upcoming as well as the Last.fm datasets with different values of N, *i.e.,* the number of extracted events, ranging from 2 to 10. Hence, for each Flickr RSS Feed, we report the average runtime of the related top N events extracted. With respect to Fig. 5, the maximum value of runtime is about 5.403(s) in the Upcoming dataset and about 4.292(s) in the Last.fm dataset, whereas the minimum value is around 3.292(s) in the Upcoming dataset and about 2.975(s) in the Last.fm dataset which is efficient and satisfiable.

5 Conclusion and Future Work

In this paper, we have tackled the challenge of event identification in social media RSS Feeds. We have formulated this task as a real-time problem and introduced a novel probabilistic approach for events mining from heterogenous social media RSS Feeds, called RssE-Miner, in order to get out these events. In particular, our approach relies on a better tradeoff between event mining accuracy and swiftness by applying the probabilistic Naive Bayes model to Flickr data. Our experiments suggest that our approach yields better performance than the baselines on which we build. To the best of our knowledge, event identification (using Naive Bayesian model) in such social media RSS Feeds have never been modeled before. In future work, we will focus on further study other more sophisticated smoothing method than Laplace to improve Naive Bayes performance. Our future research will focus also on event ontology enrichment. Indeed, from these events, we aim to enrich an event ontology. Such an ontology is useful in providing accurate, up-to-date information in response to user queries.

References

1. Aggarwal, C., Subbian, K.: Event detection in social streams. In: Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012, pp. 624–635. SIAM/Omnipress, Anaheim (2012)
2. Allan, J. (ed.): Topic Detection and Tracking: Event-Based Information Organization. Kluwer Academic Publishers (2002)
3. Baeza-Yates, R., Berthier, R.N.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
4. Becker, H., Naaman, M., Gravano, L.: Learning similarity metrics for event identification in social media. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, WSDM 2010, pp. 291–300. ACM (2010)
5. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. In: Proceedings of the 5th International Conference on Weblogs and Social Media, ICWSM 2011. The AAAI Press, Barcelona (2011)
6. Bekkerman, R., Mccallum, A., Huang, G.: Automatic categorization of email into folders:benchmark experiments on enron and sri corpora. Technical Report IR-418, University of Massachusetts, Amherst, USA (2004)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: an update. SIGKDD Explorations 11, 10–18 (2009)
8. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Doctoral thesis, The University of Waikato (April 1999)

9. Hogenboom, A., Hogenboom, F., Frasincar, F., Kaymak, U., van der Meer, O., Schouten, K.: Detecting economic events using a semantics-based pipeline. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 440–447. Springer, Heidelberg (2011)
10. Hogenboom, F., Hogenboom, A., Frasincar, F., Kaymak, U., van der Meer, O., Schouten, K., Vandic, D.: SPEED: A semantics-based pipeline for economic event detection. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 452–457. Springer, Heidelberg (2010)
11. Java, A., Finin, T., Nirenburg, S.: Semnews: A semantic news framework. In: Proceedings of the 21st National Conference on Artificial Intelligence, pp. 1939–1940. AAAI Press (2006)
12. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2004, pp. 25–29 (2004)
13. Li, R.D.W., Abdin, K., Moore, A.: Approaching real-time network traffic classification. Technical Report RR-06-12, Department of Computer Science, Queen Mary, University of London, London (2006)
14. Papka, R., Allan, J., Lavrenko, V.: On-line new event detection and tracking. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1998, pp. 37–45. ACM (1998)
15. Reuter, T., Cimiano, P., Drumond, L., Buza, K., Schmidt-Thieme, L.: Scalable event-based clustering of social media via record linkage techniques. In: Proceedings of the 5th International Conference on Weblogs and Social Media, ICWSM 2011. The AAAI Press, Barcelona (2011)
16. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 851–860. ACM, New York (2010)
17. Song, Y., Kolcz, A., Giles, C.L.: Better naive bayes classification for high-precision spam detection. Softw. Pract. Exper. 39(11), 1003–1024 (2009)
18. Yang, Y., Pierce, T., Carbonell, J.: A study on retrospective and on-line event detection. In: Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval, SIGIR 1998, pp. 28–36. ACM, New York (1998)