An Event Search Platform Using Machine Learning

Marcelo Aguiar Rodrigues
Polytechnic of Coimbra
ISEC
Coimbra, Portugal
a21180873@isec.pt

Rodrigo Rocha Silva
São Paulo State Technological College
CISUC
Mogi das Cruzes, Brasil
rodrigo.rsilva@fatec.sp.gov.br

Jorge Bernardino
Polytechnic of Coimbra - ISEC
CISUC
Coimbra, Portugal
jorge@isec.pt

***Abstract*—Currently, the evolution of technology allows to find which events occurs around us at any given location. Social networks are one of the reasons of this trend and new applications are emerging aiming at finding and disclosing events. This paper proposes a platform of event searching. In particular, we propose a new architecture that uses machine learning to classify events with tags. An experimental evaluation with different types of algorithms was done using Facebook as a source of dataset events.**

Keywords - events search; data mining; machine learning

I. INTRODUCTION

Nowadays, the quantity of digital information about what happens around us is dispersed in many applications. Usually, working with events, implies dealing with variables like date, location and time [1]. With the emergence of social networks, other types of relevant information should be considered, like a list of users that have an interest in the event or a list of users who will attend the event. In the literature, it is usually mentioned that users like sharing their stories, opinions, photos, and videos on social networks, creating a direct and social interaction between the participants on a certain event [2].

The events are a natural way to show an observable occurrence, grouping people, places, times, and activities [3]. Also, they might be considered as observable experiences that are often documented through photos and videos [4].

This paper presents a new idea of a platform for event searching. In particular, we propose a new architecture using machine learning to provide more accurate information according to the user interests. The main advantage of the platform is to bring a more personalized system where the user can find what s/he needs and get recommend events based on personalized tags that s/he follows.

Our main contributions are: a new approach for an event search platform using machine learning; integration of LODE ontology to structure event data and use it on classification; and classification tests with 101,121 events with 83.33% of classified events.

The remainder of this paper is organized as follows. Section II describes our event search platform and its architecture. Section III describes the process of organization data with the LODE ontology. Section IV describes the algorithms used and the experimental tests for events classification. Finally, Section V concludes the paper and presents future work.

II. THE EVENT SEARCH PLATFORM

Finding digital content related to events is challenging, requiring searching at different sources and sites [5] and sometimes, the data is ambiguous and incomplete.

A. The idea

The goal is to create an event search platform where every event can be classified with several tags. A good similarity is for example the Foursquare application [6]. Each place is associated to multiple tags, e.g., a restaurant can be associated to pasta, cocktails, pizza and, others, depending on their service type.

Our idea is to take advantage of these tags system and apply it on an event search platform. For example, a Bruce Springsteen concert [7] may be associated with tags like rock, hard rock or folk. Merging these two concepts (events and tags) can bring some advantages, such as:

- The platform can accommodate not only predefined events with selected tags, but all kind of events. For example, we can have one event related with music and one event related with a scheduled construction work on a specific street;
- Creation of customized lists according to the user's preferences;
- Creation of a more personalized search engine to return more accurate events to the user;
- Better interactivity with users, allowing them to create and classify events with tags. If a tag does not exist, the user can create the tag at the time of creating the event, allowing the system cover all type of events with the user input. As a business rule, each event should have at least one tag.

Machine learning is used for events classification to bring more improvements in the recommendation and search of events, as well as on the notification of events. Its main goal is to classify events obtained from APIs in several tags, but it can also help make the system more personalized to the user. For the platform, we can add a new feature like the suggestion of tags in the process of creating an event. For example, if a tag is followed by 1000 users, the suggestion of this tag at the time the event is created, can reach a larger number of people who might be interested in participating in it.

Yet, there is some concern about allowing users to create their events as well as classify them. This feature may lead to

DOI Reference Number: 10.18293/SEKE2017-159

inconsistent data and may have repercussions on the events classification. To solve this problem, we can use the recommendation system proposed in [8]: at the time of creating the event, the platform recommends a series of tags that can be used to classify the event depending on its data. If it is necessary to create a new tag, the submitted event must go through an approval process, to verify that the tags created are related with the event. This way, we think that it is possible to solve the issue of data inconsistency generated by the user.

B. *Proposed architecture*

Figure 1 shows the architecture of our platform and how its components communicate between them. Next, we will describe every component and its main function.

Client Applications are the applications that allow the user interact with the system and view the lists of events as well as create their own events. These applications will communicate with the server through the developed API, which is based on HTTPS (Hypertext Transfer Protocol Secure) protocol. The data sent is in JSON (JavaScript Object Notation) format and consists of event data.

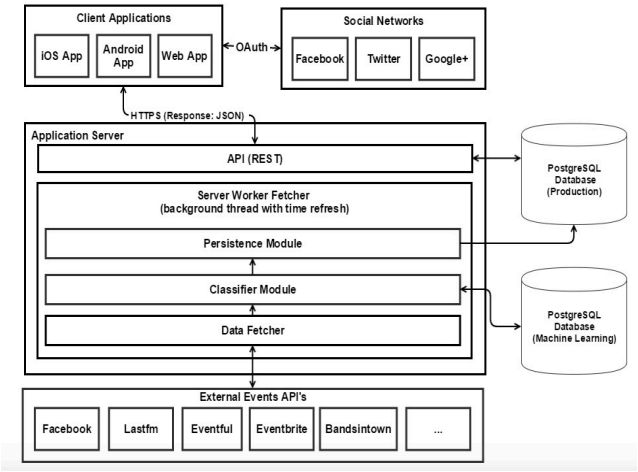

Figure 1. Proposed architecture

Application Server is the first responsible for supporting client applications, containing a RESTful API [9] to handle the requests about events. There is also a module for managing event classification (Server Worker Fetcher).

In order to save the data, the current architecture provides two databases. The PostgreSQL (Production) database will be the database that stores all event data already classified and used by the applications described above. The PostgreSQL (Machine Learning) is a copy of PostgreSQL (Production) database and will only be used as training base of the algorithm to classify the events coming from external APIs. This database will be updated periodically to improve event classification.

Server Worker Fetcher is a server worker whose main function is to get and classify events. It is divided into three modules: Data Fetcher module handles the communication between external APIs and the server to get events data; Classifier module handles the classification of event data through a machine learning algorithm. The PostgreSQL (Machine Learning) database provides the data for classification; Persistence Module stores event data already classified into the PostgreSQL (Production) database.

Finally, the external APIs are the APIs responsible to provide events.

III. EVENT DATA WITH LODE ONTOLOGY

One of the main contributions of this work is to show a different approach of an event search platform using machine learning. This section aims to present the process of integrating the LODE ontology to structure the event data and use them in the classification. A comparative study of several APIs was carried out, to understand which entities are similar between them.

The purpose of LODE ontology is to enable interoperable modelling of factual aspects to encapsulate the most useful properties to describe events [3]. The goal is to give answers about *"What is happening?"*, *"Where it is happening?"*, *"When it is happening?"*, *"Who is involved?"* [3], and organize this information in several properties, which are *Event*, *atPlace*, *atTime* and, *involved*.

Events often need a response from the user. This response is called R.S.V.P, which means "Répondez S'ill Vous Plaît" in French. This data permits to know if whether users will attend the event. This status is represented in several users counts that can be subdivided into the following categories:

- Attending guests: represents the guests that will attend the event;
- Declined guests: represents the guests that won't be attending the event;
- Interested guests: represents the guests that have interest in the event but don't know if they will be attending;
- No reply guests: guests who didn't reply to the invite;
- Maybe guests: guests that maybe will attend the event.

The final attributes of our dataset can be seen on Table I:

TABLE I. *Attributes of our dataset*

Properties	Attributes
atPlace	venue_latitude
	venue_longitude
atTime	event_start_hour
	event_end_hour
	event_start_day_of_month
	event_end_day_of_month
inSpace	venue_id
involved	artist_id
social	event_attending_count
	event_declined_count
	event_interested_count
	event_noreply_count
	event_maybe_count

IV. EXPERIMENTAL EVALUATION

In the experimental evaluation, we intend to find the best classification result in order to validate events classification for only one tag.

A. Algorithms

We use the following algorithms provided by *Weka,* corresponding to different classification categories: Decision Trees, was chosen the Random Forest [10], for the lazy classifiers, the K-Nearest Neighbors [11] was chosen, whose implementation in Weka is named IBk and, for function classifiers, Sequential Minimal Optimization (SMO) [12] was chosen, an algorithm for training support vector machines.

B. Classification results and discussion

In order to perform these experiments, it was necessary to create two datasets. The first dataset has about 1,121 events sourced from Facebook. The second dataset is a generated dataset with about 100,000 events.

The tests performed in this work are evaluated using the correctly classified instances. The 10-fold cross validation test mode was used, which means that 90% of the data is used for training and 10% for testing in each fold test.

The first test aimed to get the classification results for both datasets, to understand the first results, without changing the data as well as the algorithms. Table II shows the difference between the results obtained for the dataset with Facebook events in relation to the randomly dataset.

TABLE II. *Results of the first classification test*

	% of correctly classified instances	
Algorithms	***Facebook Dataset***	***Randomly Dataset***
IBk	50.02%	100%
SMO	46.67%	100%
Random Forest	70.28%	100%

For the IBk and SMO algorithms, the difference is around 50% and for the Random Forest algorithm the difference is around 30%. These differences are related to some missing values in the Facebook dataset. Since the dataset is composed by numeric data, the APIs do not always return all data to the attributes, leading to missing values. These same values are represented as zero, which on our view, affects the classification of events. There are three approaches to lead with missing values: mark, impute and remove missing values.

The technique to mark missing values aims to change the missing data that will be represented as "?". Yet, instances with missing values do not have to be removed and we can replace the missing values with other values with the mean of the numerical distribution. In order to have also missing data on our generated dataset, we created another one and we did the same tests for this new dataset. The results for these two techniques described above can be seen on Table III.

Comparing the results of Table II with Table III for the generated dataset, we can conclude that adding values missing also made the results worse. It is clear that both approaches cannot be taken into account in the classification process.

The last approach to deal with missing values is to remove events that contain one or more attributes with missing data. Considering the results of the previous Table II and III, we chose Feature Selection to understand which attributes are the most useful or relevant to our scenario. This is important because the number of attributes used can make the work of the classifier more difficult, making it slower and even diminishing accuracy.

TABLE III. *Classification results for mark and impute missing values techniques*

Algorithms	**Technique**	**% of correctly classified instances**	
		Facebook Dataset	***Randomly Dataset***
IBk	Mark Missing Values	41.60%	62.09%
	Impute Missing Values	48.12%	68.88%
SMO	Mark Missing Values	43.86%	77.96%
	Impute Missing Values	43.89%	76.33%
Random Forest	Mark Missing Values	61.54%	70.45%
	Impute Missing Values	68.89%	79.44%

Feature selection method aids to create an accurate predictive model. They help choose features that will give good or better accuracy whilst requiring less data [13]. They can be used to identify and remove irrelevant or redundant attributes from data that do not contribute to the accuracy of a predictive model or can decrease the accuracy.

Many feature selection techniques are supported in Weka. We choose the Information Gain Based Feature Selection, a popular technique to calculate the information gain based on the entropy concept. It is used as a measure of feature relevance in filter strategies that evaluate a feature individually [14]. We can calculate the information gain for each attribute for the output variable. Entry values vary from 0 (no information) to 1 (maximum information). Those attributes with more information will have a higher information gain than the others. Since the Facebook dataset represents the actual data of our platform, we only applied this technique on this dataset to understand the most relevant attributes. Table IV only shows the attributes that have a contribution for our case.

TABLE IV. *Attributes contribution gain results*

Attributes	**Information Gain**
artist_id	0.8864
event_start_hour	0.4246
event_end_day_of_month	0.4246
venue_longitude	0.3639
event_maybe_count	0.1003
event_interested_count	0.0600
event_attending_count	0.0423
venue_id	0.0403

We used an arbitrary cut-off of 0, which means that the attributes with this value were removed from the dataset. We proceeded again to the classification tests with the changes made on the dataset. The results can be seen on Table V.

Table V shows a great improvement comparing with results of Table II. Random Forest increased 13.05%, IBk increased 27.02%, and SMO increased 23.07%. This feature selection

showed that we have a lot of irrelevant attributes making the classifiers slower and even in some cases diminishing its accuracy.

TABLE V. *Classification results after apply the Information Gain Feature Selection*

Algorithms	% of correctly classified instances
IBk	77.74%
SMO	69.74%
Random Forest	83.33%

In conclusion, given the large difference in the results between Table II and Table III, compared with Table V, it is possible to verify that one of the problems of our dataset and the unsatisfactory results of the first two tests are related with the missing data. However, with the use of Information Gain feature selection technique, when classifying with only the most relevant attributes of our dataset, even with missing data, the results have risen considerably. In addition, within the relevant attributes it is possible to observe that 3 attributes are related with R.S.V.P, confirming that they bring relevant data in the classification of events.

In a first phase, feature selection needs to be applied since it allows to remove immediately the redundancy and irrelevance of some attributes. Even for a large database with 100,000 events, if we don't have missing data, the results are very good, as shown in Table II, but if we add missing data the results are worst. In this case, techniques of remove missing values should be applied, to understand the impact of these missing data in the dataset.

V. CONCLUSIONS AND FUTURE WORK

We propose an event search platform with a new architecture using machine learning. The use of machine learning aims to classify events in a specific tag. Our idea takes advantage of a tags system to agglomerate, not only predefined events on some categories, but all kind of events. Other advantages of our proposal, are to create customized data according to the user's preferences and a better interactivity between the users and events.

The LODE ontology was used to organize the data obtained from external APIs and was made an experimental study to find the best classification result and algorithm to validate the addition of machine learning on the architecture proposed. For performing these experiments, it was necessary to create two datasets: the first dataset has about 1,121 events sourced from Facebook and the second dataset is a generated dataset with about 100,000 events.

From the three algorithms used (Random Forest, IBk, and SMO), the first results weren't satisfactory for the Facebook dataset. The best result was 70.28% for the Random Forest algorithm. But, for the generated dataset, the results were good, reaching 100% of classified instances.

From the experimental tests, it was verified that sometimes the APIs return missing values which leads to a poor classification of the algorithms. Using the feature selection technique, we came to the conclusion that certain attributes of the Facebook dataset were irrelevant. After being eliminated, Random Forest obtained the best classification result, reaching 83.33% of classified instances. Comparing the results of the generated dataset in the beginning of tests with this result, it is possible to conclude that our training data can't have missing values because, the algorithms performance is worst

Although the classification result (83.33%) was good, there are open issues that we will be performed as future work. The experimental dataset has a small event base, so it is necessary to have more events to confirm the results obtained in these tests. With more events, other techniques of feature selection, such as, learner feature selection or correlation feature selection, must be considered, to understand the data generated in the dataset, to find a pattern that allows obtaining the best percentage for the classification of events.

All these results prove that the proposed platform is viable. Yet, allowing users to create and sort their events, the ambiguity and inconsistency of the data may be a problem in the future. Despite the proposals presented in this paper to solve the problem, they must be validated. Also, it is also necessary to find a solution to merge the data coming from external APIs, since each API has its own data structure. These issues lead us to other relevant issues about the performance, such as: the time that takes to build our training base and prepare the data for classifications, the performance of a classification procedure and the combination of data among the external APIs.

REFERENCES

[1] Costa-Dasilva, Ignacio, J., Gómez-Rodríguez, A., González-Moreno, J.C. and Ramos-Valcárcel, D. (2015) 'A located and user personalized event's dissemination platform', Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 28(1), pp. 71–81.

[2] Baruah, T.D. (2012), "Effectiveness of Social Media as a tool of communication and its potential for technology enabled connections: A micro-level study", International Journal of Scientific and Research Publications, Volume 2, Issue 5.

[3] Shaw, R., Troncy, R. and Hardman, L. (2009) 'LODE: Linking open descriptions of events', in Lecture Notes in Computer Science. Springer Nature, pp. 153–167.

[4] Troncy, R., Fialho, A., EURECOM, Hardman, L. and Saathoff, C. (2010) 'Experiencing events through user-generated media'

[5] Girolami, M., Chessa, S. and Caruso, A. (2015) 'On service discovery in mobile social networks', Computer Networks: The International Journal of Computer and Telecommunications Networking, 88(C), pp. 51–71.

[6] Made, L. and SFfourSq (2016) Food, Nightlife, entertainment. Available at: https://foursquare.com/ (Accessed: 25 November 2016).

[7] Springsteen, B. (no date) Official Website. Available at: http:// brucespringsteen.net (Accessed: 22 November 2016).

[8] Ricci, F., Rokach, L. and Shapira, B. (eds.) (2015) Recommender systems handbook. Springer Nature.

[9] Garriga, M., Mateos, C., Flores, A., Cechich, A. and Zunino, A. (2016) 'RESTful service composition at a glance: A survey', Journal of Network and Computer Applications, 60, pp. 32–53.

[10] Breiman, L. (2001) Machine Learning, 45(1), pp. 5–32.

[11] Aha, D. W., Kibler, D. and Albert, M. K. (1991) 'Instance-based learning algorithms', Machine Learning, 6(1), pp. 37–66.

[12] Cohen, (1995), "Fast Effective Rule Induction," In Proceedings of the Twelfth International Conference on Machine Learning.

[13] Guyon, I. and Elisseeff, A. (2003) 'An introduction to variable and feature selection', The Journal of Machine Learning Research, 3, pp. 1157–1182.

[14] Yang, Y. and Pedersen, J. O. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the 14th International Conference on Machine Learning. San Francisco, CA, USA, pp. 412–420, 1997