ISSN 1870-4069

Automatic Detection and Registration of Events by Analyzing Email Content

Omar Juárez Gambino[1], José-David Ortega-Pacheco[1],
Consuelo-Varinia García-Mendoza[1], Miguel Felix-Mata[2]

[1] Instituto Politécnico Nacional, ESCOM, Mexico City,
Mexico

[2] Instituto Politécnico Nacional, UPIITA, Mexico City,
Mexico

omarjg82@gmail.com, david82d@hotmail.com, cvgarcia@ipn.mx, mmatar@ipn.mx

Abstract. In this paper we describe a system for automatic event detection and registration. This system monitors the received emails by the user and determines which of them are events based on their content. For analyzing the email content, Natural Language Processing techniques were used. Once an email is classified as an appointment, the details about this event are extracted and registered in the calendar of the user. Besides a mobile application notifies the user about this event and allows to view the specifics. 300 emails written in Spanish were collected to evaluate the performance of the system. A Naïve Bayes classifier was used to identify emails containing events and the specifics were extracted using pattern matching showing an overall system accuracy of 87%.

Keywords: Event detection, text classification, information retrieval.

1 Introduction

Electronic media has changed the way people communicate. Nowadays it is possible to send and receive a message from a person who is on the other side of the world in seconds or even have a conversation in real time. One of the most popular means of communication is email and it remains as the most pervasive form of communication in the business world. According to [3] in 2014 a business person receives and sends an average of 121 email per day. Due to the large amount of information users receive through email, the time invested in reviewing this information is increasing. In addition much of this information is impersonal, spam or little relevant to the user.

Text classification is an Artificial Intelligence task for automatically assigning predefined categories based on text content [8]. A comprehensive study of methods and applications of text classification is found in [1]. One interesting application is email classification, this problem has been approached from different perspectives. In [12] a Bayesian network was used to classify emails between

pp. 35–43; rec. 2016-05-28; acc. 2016-09-15

spam and non-spam. The authors show that some specific domain attributes like mail subject, email domain and typical grammatical structure of spam mails help to improve the precision of the classification. Assigning messages to user-created folders is another interesting problem, in [7] the authors used the Enron corpus [6], which is a large set of email messages made public, to prove the SVM classifier for this task.

Event detection is another application that analyzes text contents, but the objective is to determine if there is an event specified into it [4]. This application is specialy useful for automatic email analysis. Nowadays some email services can detect and even register (in the user calendar) some events as flight departures and concerts attendance but emails must have a specific structure. For unstructured information [2] proposed algorithms for isolating events emails from incoming messages written in English and algorithms for extracting specific information. We have not found yet an integrated system for automatically detect and register events in unstructured emails written in Spanish.

In this paper we describe the development of a system that classifies emails written in Spanish as an event or non-event and extracts the specifics for later register it in the user calendar. Natural Language Processing techniques were used to process the text and a Naïve Bayes algorithm was used for classification. In the following section we present an overview of the system and a description of each implemented phases (Section 2); the experimentation and results (Section 3); and finally our conclusions and future work (Section 4).

2 System Description

The developed system is composed by several subsystems and modules. In Figure 1 we show an overview. The whole system includes the following:

- Interaction with web email services. Gmail® and Outlook® are two of the most popular web email services, from these services we got the emails received by the user.
- Server side application. The received emails are processed by this application in order to determine if their content is related to an event.
- Interaction with a web application. If an event is detected the information related is registered in Google Calendar®.
- Mobile application. A mobile application receives a notification when a new event is registered and the user can display the specific details of it.

One of the main objectives in this work was to integrate the full process in a system; from the arrival of a new email, to the user's notification of the registered event. Despite the system has different modules, the user interacts only with the mobile application making the rest of the components invisible. All the components are described next.

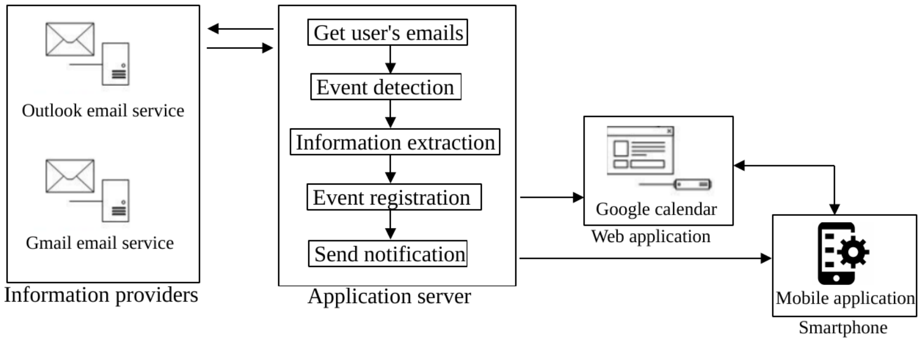


Fig. 1. System overview.

2.1 Email Client

The first module of the system is used for obtaining the received emails by the user, for this purpose an email client was developed. The client is constantly monitoring the user inbox and when a new one arrives the content is sent to the event detection module. To get the received emails, the address and the password of the account is required, this information is provided by the user during the configuration of the mobile application. Due to the sensitivity of the information, this is encrypted using the MD5 algorithm [11].

2.2 Text Processing

Before the system can determine if an email contains an event description, it needs to be processed. The email received by the client is in HTML format therefore a parser is used to extract the content. Once the content is obtained several Natural Language Processing techniques were applied using the Freeling [10] language suite:

1. Tokenization. Separates the content into words.
2. Lemmatization. Transforms the original words into their base form (lemma)
3. Named Entity Recognition (NER). Identifies entities like proper names and places. For our purpose dates and email addresses were also recognised.
4. Stopwords removal. A list of not useful words were eliminated, like pronouns and articles.

As an example of text processing consider the following email:

- "Hola buen día. Por medio de la presente se le informa que deberá asistir a la reunión de trabajo con el propósito de firmar su contrato laboral. La cita será el día miércoles 27 de Agosto de 2016 a las diez de la mañana en el World Trade Center de la Ciudad de México. Se le pide puntualidad. Cualquier aclaración comunicarse con nosotros. Atte. Luis Antonio Hernández De La Luz. Director General de COMPUMEX S.A. de C.V." (*Hello good morning.*

Through this you are advised to attend a working meeting with the purpose of signing the employment contract. The appointment will be on Wednesday, August 27, 2016 at ten o'clock in the World Trade Center of Mexico City. You are asked to be punctual. Contact us for any further information. Sincerely Luis Antonio Hernández De La Luz. COMPUMEX CEO).

After the text processing is applied the generated output is:

- "hola bueno día por_medio_de presente informar deber asistir reunión trabajo propósito firmar contrato laboral cita ser día date_pattern np_pattern se pedir puntualidad cualquiera aclaración comunicar np_pattern np_pattern" (*Hello good day through_this inform ask asist work meet purpose sign employment contract appointment be day date_pattern np_pattern you be ask punctual contact any further information np_pattern np_pattern*).

The generated output shows some named entities marked as "date_pattern" and "np_pattern". Some of these entities will be used to extract the details of the appointment. It is important to mention that all these transformations are used only during the appointment detection phase, so the original version of the email is preserved for the information extraction phase.

2.3 Event Detection

The processed text is sent to a Naïve Bayes classifier to determine if the email is related to an event or not. The classifier was previously trained with examples of both classes and provides the probability of belonging to classes.

2.4 Information Extraction

Once an email has been classified as an event it is important to extract its details. For this objective we follow a pattern matching approach. To correctly extract the details, the email content needs to have the following characteristics:

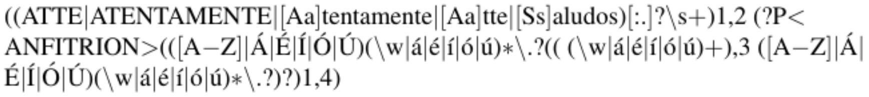
1. Written in formal style. Using appropiate language, avoiding the use of slang, abbreviations, etc.
2. Lexically and grammatically correct. Free of lexical and grammatical errors.
3. No nested references. The details of event must be contained in the current email, not in a previously referred.

The patterns were defined experimentally using the most common grammatical structures in formal emails. The content of the email is compared with the patterns, looking for matches. As an example, by using the patterns with the email shown in the text processing section, the extracted information will be:

- Purpose: la reunión de trabajo con el propósito de firmar su contrato laboral (*a working meeting with the purpose of signing the employment contract*)
- Starting date: 2016-08-27 10:00 (YYYY-MM-DD HH:MM)
- Ending date : 2016-08-27 10:00 (YYYY-MM-DD HH:MM)

- Place: el Word Trade Center de la Ciudad de México (*the World Trade Center of Mexico City*)
- Host: Luis Antonio Hernández De La Luz

In Figure 2 we show the pattern used for host extraction, considering the most common variations according to our corpus. It is likely that not all the emails classified as events will contain the full details. To handle this the system takes some considerations, for instance, if the purpose of the event is not found in the email content, the subject of the email is used instead. If the ending date is not found the starting date is used (like in the previous example). In the worst case some details could be blank, but the system would try to extract as much information as possible.



```
((ATTE|ATENTAMENTE|[Aa]tentamente|[Aa]tte|[Ss]aludos)[:.]?\s+)1,2 (?P<
ANFITRION>(([A−Z]|Á|É|Í|Ó|Ú)(\w|á|é|í|ó|ú)∗\.?(( (\w|á|é|í|ó|ú)+),3 ([A−Z]|Á|
É|Í|Ó|Ú)(\w|á|é|í|ó|ú)∗\.?)?)1,4)
```

Fig. 2. Pattern used for host extraction.

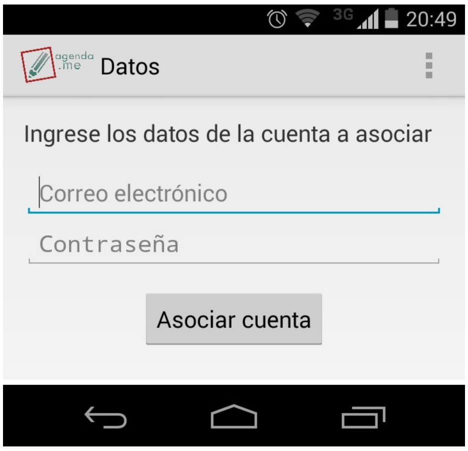
2.5 Event Registration

Calendar applications allow users to register events, thus they can remember them later. Nowadays there are a lot of calendar application available, and one of the most popular is Google Calendar. This application provided by Google associates a Gmail account to a calendar service, therefore every Gmail user has a Google Calendar. All the information related to an event that the system was able to extract it is sent to the Google Calendar web application in order to register it.

2.6 Mobile Application

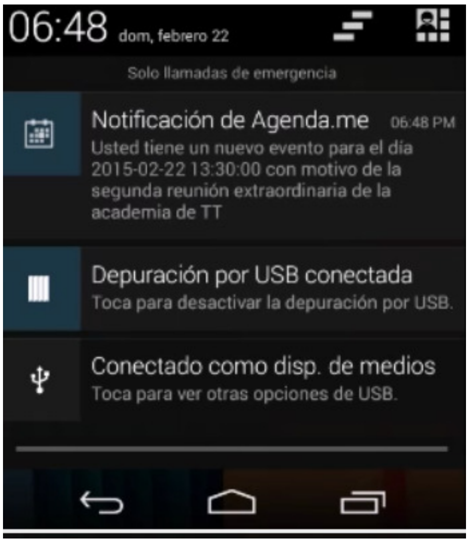
All the previous processes are invisible to the user because they are running in the server side. To allow the interaction of the user with the system an Android mobile application was developed. The user must have an active Gmail account and also an optional Outlook account. After installing the mobile application, it is necessary a configuration step. During configuration the user needs to register their email accounts (or account) by providing the address and password (see Figure 3).

When the server side application registers a new event in the Google Calendar account of the user a notification is sent to the mobile application. The user can display the details of the registered event by selecting it using the calendar of the mobile application. In Figure 4 a notification message is shown, while Figure 5 shows the calendar with the registered appointments.



Ingrese los datos de la cuenta a asociar

Fig. 3. The registration screen asking for user's email address and password.



Notificación de Agenda.me 06:48 PM
Usted tiene un nuevo evento para el día 2015-02-22 13:30:00 con motivo de la segunda reunión extraordinaria de la academia de TT

Depuración por USB conectada
Toca para desactivar la depuración por USB.

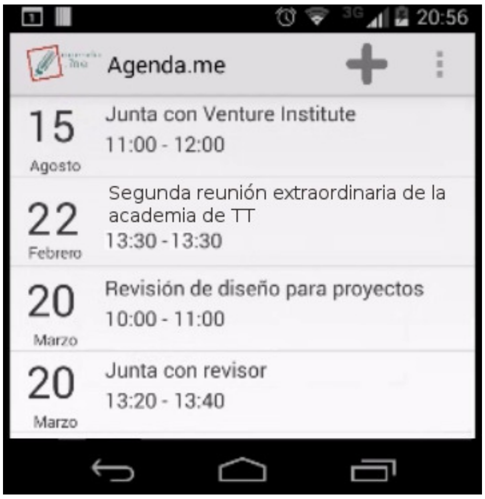
Conectado como disp. de medios
Toca para ver otras opciones de USB.

Fig. 4. The first element of the list shows a notification of a new event (*You have a new event in 2015-02-22 13:30 00 with the purpose of the second extraordinary meeting of the TT academy*).

3 Experiments and Results

3.1 Corpus

A corpus of Spanish emails of events and non-events was used to test the system. To our knowledge there is not a publicly available corpus with these characteristics, therefore it was created by ourselves. During a two months period



20:56

Agenda.me

15 Agosto — Junta con Venture Institute 11:00 - 12:00

22 Febrero — Segunda reunión extraordinaria de la academia de TT 13:30 - 13:30

20 Marzo — Revisión de diseño para proyectos 10:00 - 11:00

20 Marzo — Junta con revisor 13:20 - 13:40

Fig. 5. The calendar shows some registered appointments, the second element is the previously notified event.

500 emails were collected from 30 different accounts. After that, two annotators manually annotated the emails as event and non-event. We discarded 26 emails because annotators could not reach consensus. Finally, we manually selected 150 event emails that met all the characteristics described in the information extraction section and 150 non-event email in order to have a balanced corpus.

3.2 Classification Test

For classifying the emails, the system used a Naïve Bayes algorithm. Following a vector space model all the words in the processed emails are counted and represented as a vector of frequencies. These vectors are the features that the Naïve Bayes classifier will use during training and testing. To evaluate the accuracy of the system a 10-fold cross validation was used. In Table 1 we show the average accuracy obtained for both classes.

Table 1. Average accuracy for event and non-event classes.

Class	Accuracy
Event	94%
Non-Event	97%

As we can see the system obtained more than 90% of correctly classified emails for both classes. This results show that it is possible to identify 9 of every 10 emails correctly using only their content.

3.3 Information Extraction Test

The system applies an extra phase to the emails classified as event for extracting the important information. If this information is not correctly extracted the event registered on the user's calendar will not be useful. For evaluating this module, we performed 3 different tests over 20 random selected event emails. The extracted information of every test email was compared with the manually extracted information described in the information extraction section. In Table 2 the obtained accuracy for every test is shown.

Table 2. Results for the information extraction module.

Test	Accuracy
1	78%
2	85%
3	81%

The results in Table 2 show that the automatically extracted information has an average of 81% of accuracy. Finally, in Table 3 we show the overall system accuracy by averaging the results obtained in both modules.

Table 3. Overall system accuracy.

Module	Accuracy
Classification (only event emails)	94%
Information extraction (3 test average)	81%
Overall system	87%

4 Conclusions and Future Work

In this work a system for automatically identify emails containing information regarding to an event was developed. The system can also extracts the details of the event and registers this information in a calendar associated with the user. Besides, a mobile application was created to notify the user when a new event is detected in the inbox and by using this application the user can view the specific details of the registered event. The evaluations performed over the classification module and the information extraction module got an overall system accuracy of 87%. The results show that it is possible to integrate different modules in a system to fulfill this task with good accuracy. As a future work, we would try other classifiers like SVM and MaxEnt which has proven effective in text classification task [5, 9]. It would be useful to increase the size of the corpus and also include a spell correction module to handle some common writing errors. For the information extraction module, more experiments are required in order to improve the accuracy, a richer lexical and syntactical representation could help.

Acknowledgments. We thank the support of Insituto Politécnico Nacional (IPN), ESCOM-IPN, UPIITA-IPN, SIP-IPN projects numbers: 20160815, 20160819, COFAA-IPN and EDI-IPN.

References

1. Aggarwal, C., Zhai, C.: A Survey of Text Classification Algorithms. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 163–222. Springer US (2012)
2. Black, J.A., Ranjan, N.: Automated event extraction from email (2004)
3. Group, R.: Email Statistics Report, 2014-2018. http://www.radicati.com/wp/wp-content/uploads/2014/01/Email-Statistics-Report-2014-2018-Executive-Summary.pdf (Jun 2014)
4. Hogenboom, F., Frasincar, F., Kaymak, U., Jong, F.D.: An overview of event extraction from text. In: Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011). vol. 779, pp. 48–57 (2011)
5. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: ECML-98, 10th European Conference on Machine Learning. pp. 137–142 (1998)
6. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML. Lecture Notes in Computer Science, vol. 3201, pp. 217–226. Springer (2004)
7. Klimt, B., Yang, Y.: Introducing the Enron corpus. In: CEAS (2004)
8. Korde, V., Mahender, C.N.: Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications 3(2), 85 (2012)
9. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: IJCAI-99 Workshop on Machine Learning for Information Filtering (1999)
10. Padró, L., Stanilovsky, E.: FreeLing 3.0: Towards Wider Multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012). ELRA, Istanbul, Turkey (May 2012)
11. Rivest, R.L.: The MD5 Message-Digest Algorithm. Request for Comments (RFC) 1321 (April 1992)
12. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: AAAI-98 Workshop on Learning for Text Categorization. pp. 55–62 (1998)