I.J. Information Technology and Computer Science, **2015, 09, 15-22**
Published Online August 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.09.03

Entity Extraction from Business Emails

Juan Li
North Dakota State University, Computer Science Department, Fargo, 58078, USA
E-mail: j.li@ndsu.edu

Souvik Sen
North Dakota State University, Computer Science Department, Fargo, 58078, USA
E-mail: souvik.sen @ndsu.edu

Nazia Zaman
North Dakota State University, Computer Science Department, Fargo, 58078, USA
E-mail: nazia.zaman @ndsu.edu

***Abstract*—**Email still plays an important role in today's business communication thanks to its simplicity, flexibility, low cost, and compatibility of diversified types of information. However processing the large amount of emails received consumes tremendous time and human power for a business. In order to quickly deciphering information and locate business-related information from emails received from a business, a computerized solution is required. In this paper, we have proposed a comprehensive mechanism to extract important information from emails. The proposed solution integrates semantic web technology with natural language processing and information retrieval. It enables automatic extraction of important entities from an email and makes batch processing of business emails efficient. The proposed mechanism has been used in a Transportation company.

***Index Terms*—**Email, entity extraction, natural language processing.

I. Introduction

Over the past five decades, email has become one of the easiest and reliable modes of communication, mainly because of its efficiency, low cost and support for wide range of information [1]. Recent studies show that email is still number one online activity though there are new concepts like social networking [3]. Corporate users send and receive about 110 messages [5] per day in average and out of them one third are messages sent. Those statistics are quite constant and has not been changed too much in last decade [6]. According to a report [7], about 80% of the business users prefer email communication over others for their work purpose. While another report [8] says that 62% of the employees in United States can be considered as Networked Workers as they use Internet and email on their work on daily basis.

Information generated by business entities can be considered as highly useful asset based on how well it is managed. Email is not different here [9]. Email is now essential for many of the common industrial [9, 10, 11] functions such as task management, collaboration, generating alerts, archiving and interoperability. It is pretty common for many of the organizations to receive product or service requests via email. To process the requests, employees of the organization have to read the emails and manually extracted important information from the emails. Normally, a company may receive thousands of such business emails every day. Therefore, to process emails quickly, an automated system is essential. This automatic processing will extract and store the featured information to provide the necessary business service.

For example, a freight company provides trucking and freight services for both residential and commercial shippers. Although they provide a web page for shippers to register their fright, they still receive thousands of freight shipping requests by emails every day. Therefore, it is important for the company to serve these email requests promptly to ensure their freight gets to its destination safely and on time. To serve the clients better and faster, the company needs to know the details about the request such as freight size, location, destination, and timeline. An email information extraction program should extract all of these important information as correctly as possible and save the extracted information in the company database for further service. Time is another issue here. Manual reading emails and providing service can take a long time and user can suffer because of that. Hence an automatic email extraction procedure will surely solve that time delay issue. Many ecommerce companies record email receipts of online transactions which are full of essential product information including product category, price, date of purchase etc. If this information can be extracted and saved in a good manner, it can be used for several purposes including a recommendation system [2]. If the system can identify the type of product a specific user is buying, then the system can suggest further products to that user using extracted information.

In this paper, we propose an effective entity extraction mechanism to locate and retrieve important information from business emails. The retrieved entities can be utilized for business management solutions to make

Copyright © 2015 MECS

business processes more efficient, effective, and predictable. The proposed work integrates rich semantics, text mining machine learning, and natural language processing technologies.

The rest of the paper is organized as follows. In Section II, we describe the details of our methodologies. In Section III, we evaluate the proposed methods and show the effectiveness of this model with a set of experiments. Related work and concluding remarks are provided in Sections IV and V, respectively.

II. System Design

A. Overivew

The content of business email is normally different from general text data in many documents. As pointed out by Tang et al. [22], emails are often much shorter and more briefly written compared with documents such as stories and user manuals. In addition, emails often contain some faddish words or abbreviations that may not appear in traditional dictionaries. Moreover, business emails also include domain specific terms/jargons. Furthermore, besides textual data, attachment of business emails also contains very important information which should not be ignored. Standard natural language processing and text mining techniques may not be effective when they are applied to business email mining tasks.

To address the aforementioned challenges, we propose a domain knowledge-assisted information extraction mechanism to retrieve important information from business emails. Prior to the inception of email information extraction and subsequent processing, it is essential to acquire a concrete domain knowledge which captures specific information about the business. This domain knowledge can be used in direct the entity extraction process and also in creating rules to extract information. In our work, the domain knowledge is encoded as ontology, which is represented as OWL/RDF format [18]. Besides entities defined in an ontologies, named entities and noun phrases referring to specific individuals like persons, organizations, location, date, and time are generally important regardless of domains. Therefore, we should locate and extract these named entities as well.

An email contains multiple parts as shown in Fig. 1. We are processing each module at a time to extract important information from each of them. The process flow for the entire system has been depicted in Fig. 2. Before extracting important entities, we first extract information from the attachments of an email. We separate the email attachments based on their file extension such as Excel attachments, and Word attachments. As graphs and figures do not contain textual information, we do not process this kind of attachment. According to their types successive processing is performed. Attachment content can be divided into two types: unstructured content and structured content. For unstructured textual information, we simply convert it into simple text. For structured information, for example tables, we try to keep their structure. For example, we use Apache's PDFBox [29] library methods to convert the PDF files to text files.

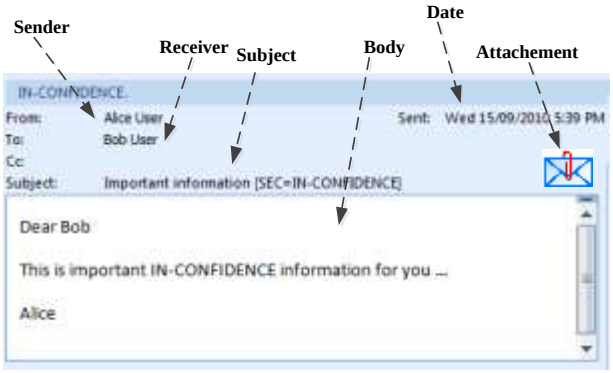

Fig. 1. Example email.

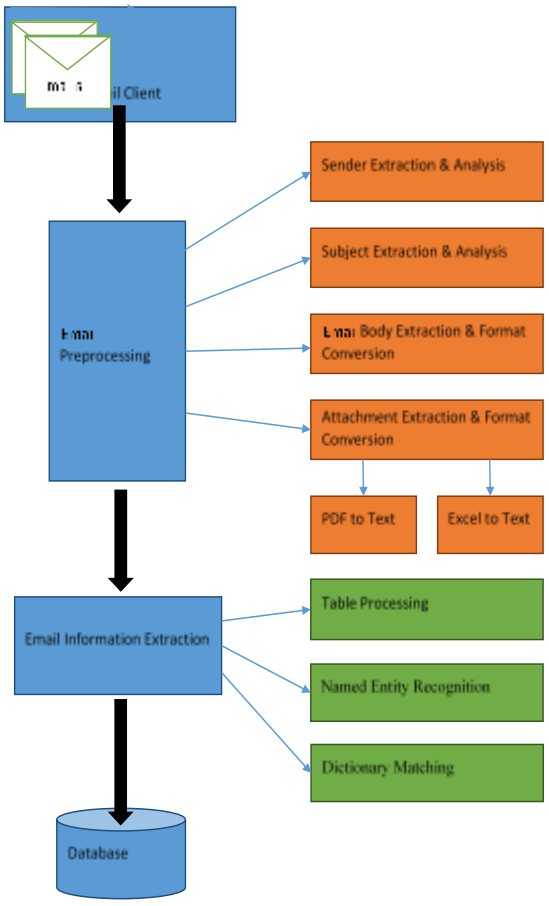

Fig. 2. System diagram for flow of the processes in the system

B. Entity Extraction from Unstrucuted Data

As most emails are written using informal natural language, therefore, as the first step, we deal with entity extraction from unstructured email text. Before extraction, we first try to remove noisy information from the unstructured text and obtain key features of the email. First, the text are cleaned from any unnecessary information such as HTML tags. And then data is segmented into sentences. We have used the Punkt

Copyright © 2015 MECS

sentence segmenter [42] to segment sentences. Then a tokenizer is used to divide text into a sequence of tokens, i.e., words in our case. We adopted the Penn Treebank Tokenization [43]. We have converted all tokens to lowercase to simplify the later semantic entity extraction process. Some of the most common, short function words, such as "the", "a", "is", "which", are useless in text analysis. We remove these words from the text to reduce the data size and improve efficiency and effectiveness of analysis. Stemming, the process of reducing a word to its root or simpler form by removing inflectional endings, is also performed in the text.

After pre-processing of the email text, we work on automatically extracting important entities from unstructured natural language. Key entities include person names, organizations, locations, dates, specialized terms and product terminology from free-form text. Existing Named Entity Extraction (NER) systems use linguistic grammar-based techniques [31], statistical models [32], i.e. machine learning, or and gazetteer based entity recognizer [23] to recognize entities. We adopt the machine leering based model – the linear chain Conditional Random Field (CRF) sequence model [44] to extract general entities. CRF (Lafferty et al., 2001) are undirected graphical models, a special case of which correspond to conditionally-trained finite state machines. Like the maximum entropy models, CRF is also based on the same exponential form, but CRF is more efficient for complete, non-greedy finite-state inference and training [44].

A CRF model is defined on observations X and random variables Y as follows:

Let $G=(V,E)$ be a graph such that
$Y=Y(Y_v)_{v\epsilon V}$, so that Y is indexed by the vertices of G.

Then (X,Y) is a conditional random field when the random variables Y_v, conditioned on X, obey the Markov property with respect to the graph:

$$p(Y_v|X,Y_w,w\neq v)= p(Y_v|X,Y_w,w\sim v)$$

where $w\sim v$ means that w and v are neighbors in G.

In this definition, a CRF is an undirected graphical model whose nodes can be divided into exactly two disjoint sets X and Y, the observed and output variables, respectively; the conditional distribution $p(Y/X)$ is then modeled.

Feature selection is very important for named entity extraction. We choose word features, such as current word, previous word, next word, and all words within a window, orthographic features, prefixes and suffixes, label sequences, and feature conjunctions.

Using the CRF model, we can extract general named entities such as persons, organizations, locations, times, etc. To effectively extract special entities defined in the domain ontology, we proposed an ontology-guided entity extraction mechanism. Although these entities are defined in the domain ontology, locating them from emails is not as easy as it appears. This is because business terms/entities used in emails tend to be information, and they may differ from what is defined in the ontology. People may use abbreviations, may write typos, and may omit word(s) from a multi-word phases in their emails.

To address the aforementioned problems, we propose a fuzzy string matching mechanism to effectively locate domain-related special entities from emails. We use the well-known string-based dissimilarity measure – edit distance to measure the distance between two strings. Edit distance is the number of operations, such as deletions, insertions, or substitutions, required to transform one string to another. It can effectively capture typographic errors, words with alternative spellings, and does not rely on the separation of word boundaries [35]. Therefore, edit distance can be applied in our system for string matching and comparing.

In this paper, we propose an effective entity extraction algorithm. In this algorithm the longest multi-word expressions that appear in the email text are mapped to the most specific concepts in the ontology. We first locate all of the noun phrases in the email, as most of the entities (class and instances) in a domain ontology are noun phrases. This noun phrase tagging process can be realized by the part-of-speech (POS) tagging [18]. For terms appeared in noun phrases, we search the semantic entities associated to the terms. Besides exact match, we also provide fuzzy search to find similar matches using edit distance. If there's a hit (i.e., exact match or edit distance smaller than a predefined threshold), we will tag the word in the email with the ontology entity ID. One word may belong to multiple ontology concepts. In such case, we tag the word with IDs of all associated semantic concepts. After we have finished the keyword-entity matching and tagging phase, we try to identify potential semantic entities in the email. This is done by scanning the tags of the terms in the same noun phrase: if multiple words in the noun phrase point to the same semantic entity, they should be considered as belonging to the same entity. The rationale of this approach is based on the observation that some words tend to be omitted and the orders of the words may be switched in phrases used in the informal emails. Through these steps, semantic entities are recognized and extracted.

C. Information Extraction from Structured Data

Many business emails include structured data. The most popular format for structured data is tabular data. Due to the lack of common schema, emails from different people or organization may use different table format. Therefore, we also need an effective strategy to extract information from the structured table data.

As shown in Algorithm 1, in the first step, we extract table header and then match the header with the domain schema. To extract those table headers, we start reading the whole text and look for a row with some header matching. After we get that header row. We start reading other rows and enter those column values under their respected column header. We continue this work until we get the end of the file or we get another table header. We use a Map to map table headers to our required

information name.

Algorithm1 Entity extraction from tables

```
Input: email m={l1, l2, …, ln},
       schema s={s1,s2, …,st},
       patterns p={p1,p2, …,pt},
       ontology o={ e1, e2, …, em},
      tableHeader TH={}
       entity set ES={}
/* li: lines, ei: ontology entity */

for each wj in l1 do
   if si=match(wi, s) then
      create tableHeader h
      h.caption=wj
      h.column=j
      insert h toTH
if TH.length>0 then
for each li after l1 in e do
    for each h in tableHeader
       create entity s
       s.type=h.caption
       s.value=w(h.colum)
       insert s to ES
else
  for each li in e do
    for each wj in li do
      if  pm=match (wj, p)
         create entity s
         s.type=pm.type
         s.value=wj
         insert s to ES
      else if et=match (wj, o)
         create entity s
         s.type=et.lable
         s.value=wj
         insert s to ES
return ES
```

It is possible that a table does not include a header. If that is the case, we use the data pattern of the domain schema and the domain ontology to match the data column. For example, to extract date format, we need to summarize all date format. Some of the commonly used date formats are represented as "Month-Date-Year", "Date-Month-Year", "Month-Date-Year", "Date/Month/ Year", "Month/Date/Year", "Month Date, Year", etc. We can use techniques such as regular expression to match such format. For columns defined in the domain ontology, we can use the fuzzy matching techniques mentioned previously to match the column data with the entity defined in the domain ontology.

III. Evaluation

The proposed system has been deployed to a freight company and been evaluated using their real emails. Fig. 3 shows the screenshot of the interface of the proposed system. We can see that entities have been automatically retrieved from the email. We have taken 2431 of emails which contains Natural language email body content, PDF Word or Excel files as attachments. We try to extract information such as freight size, location, destination, and timeline, and required vehicle type, etc.

We have used our system on 1110 emails as Natural Language text as email body, 554 emails with PDF with Text as attachment, 542 emails as PDF with tables as attachment and 225 emails as Excel files as attachment. We have counted the number of Desired Fields that appears in the email content and the number of fields that we have managed to capture. Based on the result found, we have calculated Precision and Recall. The definition of recall and precisions are defined as follows:

$$recall = \frac{|relevantEntries \cap retrievedEntries|}{|relevantEntries|}$$

$$precision = \frac{|relevantEntries \cap retrievedEntries|}{|retrievedEntries|}$$

Precision represents fraction of retrieved items which are relevant i.e. the number of correct results delivered divided by the number of all items retrieved. Recall represents fraction of relevant items that has been retrieved i.e. number of correct results achieved divided by the number of correct results that were supposed to be returned. [38].

The results are illustrated in Table 1. From Table 1, we can see that our entity extraction scheme achieves good recall and precision for natural language email and almost perfect recall and precision for tabular data.

Table1. Performance of the Entity Extraction Mechanism

Type of Content	Precision	Recall
Natural Language	86.2%	83.3%
Tabular Data	100%	96.5%

IV. Related Work

As we have discussed earlier that emails are very common medium of electronic communication for almost last 40 years, a considerable amount of research has been done on email analysis and mining to get benefit from those email data. Richardson and Domingos presented an efficient algorithm to extract product information from Emails Receipts [27]. The proposed algorithm is based on Markov Logic [26]. Markov logic is the combination or probability and logic. In their work, the authors have encountered many challenges: for example, E-receipts can be generated from different templates. Making a generalized rule is always challenging. Maximum of the E-receipts are based on plain text instead of HTML tagging and that makes the process of information extraction much more complex as data representation is

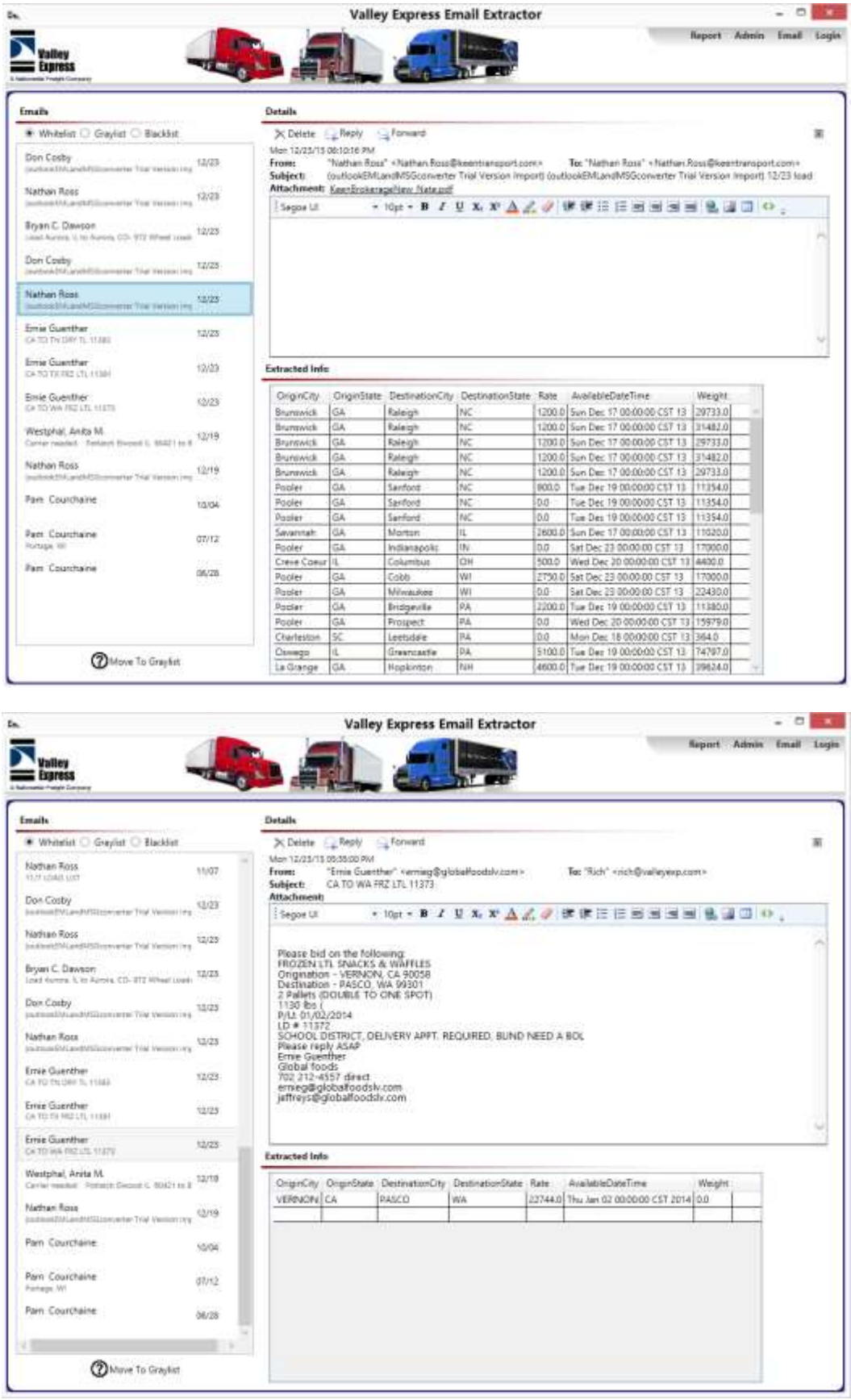

Fig. 3. Screenshot of a prototype system. (The upper figure shows the tabular data extraction and the lower figure shows the natural language data extraction)

irregular. They have created a corpus of unlabeled E-receipts and they have identified all possible templates by jointly clustering all those E-receipts [27].

In another study [28] Boufaden et al. have used semantic tagging and domain knowledge for the enterprise to extract information form an outgoing email in a company. They use the extracted data to detect the privacy risk of an organization by matching the extracted data against a set of compliance rules. Laclavík et al. present how email analysis and extraction can benefit an enterprise. They have proposed a light-weight process using various natural language processing techniques such as Named Entity Recognition (NER), Coreference Resolution (CO), Template Element Construction (TE), Template Relation Construction (TR) and Scenario Template Production (ST), then Key-Value pair based information extraction to get the important information regarding enterprise emails. The extracted information has been processed using Semantic Trees, Email Social Networks, and Graph Inference respectively. Bird et al.

Appavu et.al., proposed an classification algorithm called Ad Infinitum [39]. Ad Infinitum is an extension of the decision tree induction algorithm. This algorithm aims to classify the threatening messages in emails. For the same purpose of detecting threat emails, Shekar et al. proposed a Naïve Bayesian filter for classification of threat e-mails [40]. They applied three different Naïve Bayesian filter approaches i.e. single keywords, weighted multiple keywords and weighted multiple keywords with keyword context matching.

Stolfo presents the Email Mining Toolkit (EMT) [41], a data mining system that computes behavior profiles or models of user email accounts. These models may be used for a multitude of tasks including forensic analyses and detection tasks of value to law enforcement and intelligence agencies, as well for as other typical tasks such as virus and spam detection.

V. Conclustions

Emails are really important in our daily life as well as in the industry world. There are numerous businesses, where emails are the only way of getting information from their clients and the only way of communication with their clients. In many times, companies have to provide service through emails. Therefore, it is crucial to automatically extract all of the important information from the emails accurately.

In this paper, we propose a series of mechanisms to exact important entities from emails, especially from business emails. In particular, we first preprocess the email data. Then we utilize the domain ontology of the business to guide effective extraction. We designed different mechanism to deal with different email content format. Our mechanism integrates rich semantics, text mining machine learning, and natural language processing technologies together. The retrieved entities can be utilized for business management solutions to make business processes more efficient, effective, and predictable. We have implemented a prototype system. This system have been deployed to a freight company. Using this system can safe employee's time and energy to manually read and process emails. The performance of the proposed system has been evaluated with the email samples from the company.

There are many ongoing and future works for this project. Often emails are associated with signatures of the sender, including their name, title, address, phone number, emails etc. Sometimes email signature information create confusion in the information data extraction. It is possible that signature information gets extracted as part of the service information and that is not desired. So it is significant to remove the email signature before we start information extraction from the email. This will produce better result. Similarly, we can see quoted information from previous correspondence. This quotation should be removed as well before the entity extraction process.

Many emails contain images or other icons. As part of important information extraction, these icons and images are often not required. These unnecessary icons can be removed during the pre-processing of the emails. This will keep the information extraction process simple and easy. In some aspect, images or icons can appear into the required information that the organization wants to extract. Then processing and extracting those icons, images and storing them will be required.

References

[1] Tang, G., Pei, J., & Luk, W. S. (2013). Email mining: tasks, common techniques, and tools. Knowledge and Information Systems, 1-31.

[2] Kok, S., & Yih, W. T. (2009). Extracting product information from email receipts using markov logic. In Proceedings of the Sixth Conference on Email and Anti-Spam, Mountain View, California, USA.

[3] "Pew Internet Report: Online Activities 2010". Pew Research Center. May 2010. Web. Aug 2014. <http://tinyurl.com/ pewOnline10>

[4] "Jones, J.: Gallup: Almost All E-Mail Users Say Internet, E-Mail Have Made Lives Better." Gallup. July 2001. Web. Aug 2014. <http://tinyurl.com/Gallup01>

[5] "The Radicati Group, Inc.: Email Statistics Report, 2010." Editor: Sara Radicati. The Radicati Group Inc. 2010. Web. Aug 2014. <http://tinyurl.com/RadicatiEmail10>

[6] "Taming the Growth of Email – An ROI Analysis (White Paper)." HP, The Radicati Group, Inc. Mar 2005. Web. Sept 2014. <http://tinyurl.com/RadicatiEmail05>

[7] "80 % of Users Prefer E-Mail as Business Communication Tool." META Group Inc. 2003. Web. Sept 2014. <http://tinyurl.com/MetaEmail03>

[8] "Networked Workers. PewInternet report" Madden, M.—Jones, S. Pew Research Center.Sept 24, 2008. Web. Sept 2014. http://tinyurl.com/pewNetWrks08

[9] Laclavík, Michal, et al. "Email analysis and information extraction for enterprise benefit." Computing and informatics 30.1 (2012): 57-87.

[10] Whittaker, Steve, and Candace Sidner. "Email overload: exploring personal information management of email." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1996.

[11] Fisher, D.—Brush, A. J.—Gleave, E.—Smith, M.A.: Revisiting Whit-taker&Sidner's "Email Overload" Ten Years Later. In CSCW2006, New York ACM Press 2006.

Copyright © 2015 MECS

[12] Corbató, F. J., Merwin-Daggett, M., & Daley, R. C. (1962, May). An experimental timesharing system. In Proceedings of the May 1-3, 1962, spring joint computer conference (pp. 335-344). ACM.
[13] "Natural Language Processing." Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc. 26 October 2014. Web. 27 October 2014.
[14] http://en.wikipedia.org/wiki/Natural_language_processing
[15] "Named Entity Recognition." Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc. 26 October 2014. Web. 27 October 2014. http://en.wikipedia.org/wiki/Namedentity_recognition
[16] Zhou, GuoDong, and Jian Su. "Named entity recognition using an HMM-based chunk tagger." proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002.
[17] Cunningham, H.—Maynard, D.—Bontcheva, K.—Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL '02), Philadelphia.
[18] Fernández, Miriam, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, and Enrico Motta. "Semantically enhanced Information Retrieval: an ontology-based approach." Web Semantics: Science, Services and Agents on the World Wide Web 9, no. 4 (2011): 434-452.
[19] Cimiano, P.—Ladwig, G.—Staab, S.: Gimme' the Context: Context-Driven Automatic Semantic Annotation With C-Pankow. In WWW'05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA. ACM Press. ISBN 1-59593-046-9,2005, pp. 332–341.
[20] Laclavík, Michal, et al. "Email analysis and information extraction for enterprise benefit." Computing and informatics 30.1 (2012): 57-87.
[21] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370. <http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf>
[22] Tang, Guanting, Jian Pei, and Wo-Shun Luk. "Email mining: tasks, common techniques, and tools." Knowledge and Information Systems 41, no. 1 (2014): 1-31.
[23] "Gazetteer." Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc. 01 August 2014. Web. 20 October 2014. < http://en.wikipedia.org/wiki/Gazetteer>
[24] Aho, Alfred V, Corasick, Margaret J. (June 1975). "Efficient string matching: An aid to bibliographic search". Communications of the ACM 18 (6): 333–340. doi:10.1145/360825.360855.
[25] "Bio sequence Algorithms, spring 2005 Lecture 4: Set Matching and Aho-Corasick Algorithm." Kilpelainen, Pekka. 2005. Sept 2014. <http://www.cs.uku.fi/~kilpelai/BSA05/lectures/slides04.pdf >42
[26] M. Richardson and P. Domingos. Markov logic networks. Machine Learning, 62:107–136, 2006.
[27] Kok, Stanley, and Wen-tau Yih. "Extracting product information from email receipts using markov logic." Proceedings of the Sixth Conference on Email and Anti-Spam, Mountain View, California, USA. 2009.
[28] Boufaden, Narjes, et al. "PEEP-An Information Extraction base approach for Privacy Protection in Email." CEAS. 2005.
[29] "Apache PDFBox – A Java Pdf Library." The Apache Software Foundation. 2014. Web. Sept2014. https://pdfbox.apache.org/
[30] Wasi, Shaukat, et al. "Event Information Extraction System (EIEE): FSM vs HMM."
[31] Saleem, Ozair, Latif, Seemab. "Information Extraction from Research Papers by Data Integration and Data Validation from Multiple Header Extraction Sources." WCECS 2012,October 24-26, 2012, San Francisco, USA. http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp215-219.pdf
[32] Chiticariu, Laura, et al. "SystemT: an algebraic approach to declarative information extraction." Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010.
[33] Almgren, Magnus, and Jenny Berglund. "Information extraction of Seminar information." CS224N: Final Project (2000): 1-12.43
[34] Black, Julie A., and Nisheeth Ranjan. "Automated event extraction from email. "Final Report of CS224N/Ling237 Course in Stanford: http://nlp. stanford. edu/courses/cs224n/2004/, Spring (2004).
[35] "Apache PDFBox 1.8.6 API." The Apache Software Foundation. 2014. Web. Oct 2014.http://pdfbox.apache.org/docs/18.6/javadocs
[36] Cimiano, Philipp, Günter Ladwig, and Steffen Staab. "Gimme'the context: context-driven automatic semantic annotation with C-PANKOW." Proceedings of the 14th international conference on World Wide Web. ACM, 2005.
[37] Etzioni, O. Cafarella, M. Downey, D. Kok, S. Popescu, A. Shaked, T. Soderland, S. Weld, D. Yates, A.: Web-Scale Information Extraction in Knowitall (Preliminary Results).In WWW'04, 2004, pp. 100–110, http://doi.acm.org/10.1145/988672.988687.
[38] "Precision and Recall." Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc. 29 October 2014. Web. 31 October 2014. < http://en.wikipedia.org/wiki/Precision_and_recall>
[39] Appavu, Subramanian, Ramasamy Rajaram, M. Muthupandian, G. Athiappan, and K. S. Kashmeera. "Data mining based intelligent analysis of threatening e-mail." Knowledge-Based Systems 22, no. 5 (2009): 392-393.
[40] Shekar, DV Chandra, and S. Sagar Imambi. "Classifying and Identifying of Threats in E-mails–Using Data Mining Techniques." In Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1. 2008.
[41] Stolfo, Salvatore J., Shlomo Hershkop, Chia-Wei Hu, Wei-Jen Li, Olivier Nimeskern, and Ke Wang. "Behavior-based modeling and its application to email analysis." ACM Transactions on Internet Technology (TOIT) 6, no. 2 (2006): 187-221.
[42] Kiss, Tibor, and Jan Strunk. "Unsupervised multilingual sentence boundary detection." Computational Linguistics 32, no. 4 (2006): 485-525.
[43] "Penn Treebank Tokenization", https://catalog.ldc.upenn.edu/LDC99T42
[44] Sutton, Charles, Andrew McCallum, and Khashayar Rohanimanesh. "Dynamic conditional random fields: Factorized probabilistic models for labeling and

segmenting sequence data." The Journal of Machine Learning Research 8 (2007): 693-723.

Authors' Profiles

Juan Li received a B.S. degree from Beijing Jiaotong University, Beijing, China, in July 1997, and a Ph.D. degree from the University of British Columbia, Vancouver, Canada, in May 2008. Currently, she is an Associate Professor of Computer Science Department at the North Dakota State University, Fargo, ND, USA. Dr. Li's major research interest lies in distributed systems, including P2P networks, grid and cloud computing, mobile ad hoc network, social networking, and semantic web technologies.

Sovik Sen received a B.S. degree from West Bengal University of Technology Kolkata, India, in 2011. Currently, he is a Master student of Computer Science Department at the North Dakota State University, Fargo, ND, USA. His current research focuses on intelligent systems.

Nazia Zanman received a B.S. degree from the University of Dhaka, Dhaka, Bangladesh, in 2007, and a M.S. degree from the same university, in 2009. Currently, she is a PhD student of Computer Science Department at the North Dakota State University, Fargo, ND, USA. Her current research focuses on intelligent systems, social networking, and natural language processing.

Manuscript received February 27, 2015; May 5, 2015; accepted May 11, 2015.

How to cite this paper: Juan Li, Souvik Sen, Nazia Zaman,"Entity Extraction from Business Emails", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.9, pp.15-22, 2015. DOI: 10.5815/ijitcs.2015.09.03

Copyright © 2015 MECS