Use of Email Social Networks for Enterprise Benefit

Michal Laclavík, Štefan Dlugolinský, Marcel Kvassay, Ladislav Hluchý
Institute of Informatics, Slovak Academy of Sciences
michal.laclavik@savba.sk

Abstract

The article discusses the potential methods and benefits of the analysis of social networks hidden in the enterprise and personal email archives. A proof-of-concept prototype was developed. Social network extraction and the spreading activation algorithm are discussed and evaluated.

1. Introduction

Email communication is a source of information on personal, enterprise or community networks of an individual or an organization. Email communication analysis allows the extraction of social networks with links to people, organizations, locations, topics or time. Social Networks included in email archives are becoming increasingly valuable assets in organizations, enterprises and communities, though to date they have been little explored.

Social networks in the area of email communication have been studied to some extent. For example, communication on the Apache Web Server mailing lists and its relation to CVS activity was studied in [1]. This work also introduces the problem of identifying email users' aliases. Extracting social networks and contact information from email and the Web and combining this information is discussed in [2]. Similarly, new email clients (e.g. Postbox) or plug-in Xobni[1] try to connect email social networks with web social networks like LinkedIn or Facebook. We have also performed some experiments on the extraction of social networks from large email archives and network transformations using a semantic model [3]. Another research effort [4] exploits social networks to identify relations, and tests the proposed approaches on the Enron corpus.

We are using a similar approach to that of IBM Galaxy [5] in the Nepomuk[2] project, where the concept of multidimensional social network was introduced. In this paper we show the initial results of exploiting the email social network in order to support a better understanding of email content as well as allowing applications such as search for contact, product, service, partner or supplier within organizations or communities.

2. Information Extraction

In order to provide the social network graph hidden in the email communication, the important task is to identify objects and object properties in the text and thus to formalize the email message content and context. For object identification we use the information extraction (IE) techniques. We have developed the Ontea [6] extraction and annotation tool, which uses regular expression patterns and gazetteers[3]. These patterns and gazetteers generate key-value pairs (object type – object value). Key-value pairs are then used to build the tree and the graph of social network described in section 3. Evaluation of IE is discussed in section 4.1.

3. Email Social Network Analysis

We implemented our "social network extractor and analyzer" in Java on top of the information extraction tool Ontea and open-source graphical library JUNG.[4] The novelty of our approach is in the application of the spreading activation algorithm to the twin tasks of reconstructing the social network from emails, and then efficiently searching the social graph. The prototype implementation described below is a work in progress. The evaluation of our initial experiments is provided in section 4.

In [7] we have introduced two approaches to building multidimensional social network graphs. Here

[1] http://www.xobni.com/

[2] http://nepomuk.semanticdesktop.org/

[3] Gazetteer is simply the list of keywords (e.g. list of Spanish given names) representing an object type, which are matched against the text of emails.

[4] http://jung.sourceforge.net/

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.126

we only use the more advanced one where IE extracts complex objects (e.g. an address) consisting of simpler objects (e.g. a street name and a postal number), and preserves the information about their physical proximity by building a hierarchical tree that includes the nodes representing the sentences, paragraphs and blocks of the message in which they were found (Figure 1). Each object is linked to all the objects in all the messages in which it was found, and can have several parallel links to one object if it occurred in it in several different positions, which is our way of recording the strength of the bond.

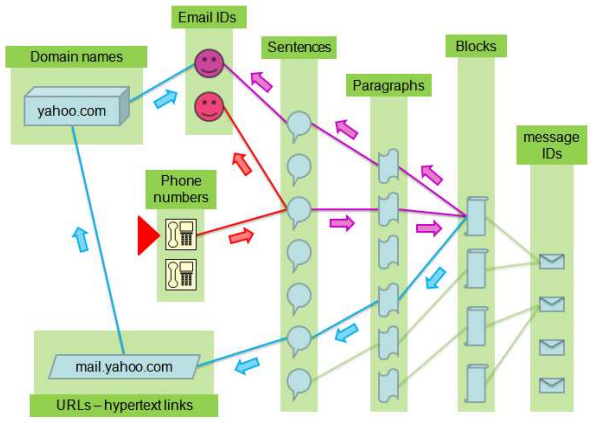

Figure 1. Spreading activation in a multipartite graph. It starts at one attribute instance (phone number) indicated by the red arrow (center left) and flows towards candidate primary entities (email IDs) by a variety of ways. Since the activation gets attenuated each time it passes through an edge, the shortest path (indicated in red) will carry over the greatest increment. But the longer paths (such as those indicated in purple and blue) can be so numerous that – depending on the value of attenuation and other parameters – their accumulated contribution may ultimately prevail

3.1. Cumulative Edge Scorer with Attenuation

In the rich tree-like structure depicted in Figure 1, it makes sense to use a more sophisticated variant of spreading activation with attenuation and activation thresholds. The structure can still be visualized as a multipartite graph (which is advantageous from the point of view of computational complexity), but the objects of each data type now require a separate partition. For the reconstruction of social network we can then use our enhanced prototype – the Cumulative Edge Scorer with Attenuation, whose algorithm is described by the following JAVA-based pseudo-code.

Breadth-first variant of Spreading Activation. `Current_wave` collection (implemented as a `HashMap` with the node as key and its `accumulated_activation` as value) contains the nodes that are going to fire. It is initialized with the attribute instance that we are trying to assign to a primary entity. In the next step, all its neighbouring nodes are placed into the `new_wave` collection as candidates for firing in the next iteration, and their accumulated activation is incremented by the value of `activation`, which reflects both the edge attenuation and the degree of the firing node. The iteration is completed when all the neighbours of all the nodes in the `current_wave` are processed. After each iteration, the number of iterations is incremented and the nodes in the `current_wave` are transferred into the `fired` collection so they do not fire again. The `current_wave` is then refilled by `new_wave.get_activated_vertices()` method that returns the nodes meeting the activation criteria. The nodes representing the primary entities never fire. They accumulate in the `new_wave` and are returned at the end of the algorithm by the `get_output_type_vertices()` method. Their accumulated activation becomes their score, and the primary entity with the highest score will "own" the attribute instance:

```
do
  for (Vertex v : current_wave)
    activation ← v.accumulated_activation();
    activation ←
      activation/(attenuation * v.degree());
    for (Edge e : v.outgoingEdges())
      Vertex w ← neighbouring_vertex(v,e);
      if (!has_fired(w))
        new_wave.smart_add(w, activation);
  total_iterations++;
  fired.add(current_wave);
  current_wave ←
        new_wave.get_activated_vertices();
  new_wave.remove_activated_vertices();
while (!done());
return  new_wave.get_output_type_vertices();
```

The resulting scores of primary entities for each attribute instance (internally stored in nested `HashMaps`) can be output in the form of a hierarchical XML. This XML can be converted to HTML by XSL transformations.

4. Evaluation

4.1. Information Extraction

We focused on the extraction of personal names and telephone numbers, which we manually annotated in each of the 50 Spanish emails to have a so-called golden standard for evaluation. We then evaluated the extraction results automatically by comparing them to manual annotations. We considered the information extraction result to be relevant, when it was strictly equal to the corresponding manual annotation. It means that both the result and the annotation must have had exactly the same position in the email text. Evaluation results are summarized in the table below.

Table 1. Evaluation of information extraction (strict match)

Type	Total relevant	Total extracted	Relevant extracted	Recall [%]	Precision [%]	F1 [%]
Person	779	788	499	64.06	63.32	63.69
Telephone	262	178	166	63.36	93.26	75.45
Fax	139	127	121	87.05	95.28	90.98

As we can see in Table 1, the results except for the fax number and telephone number extraction are not high. This is due to strict comparison of the extraction results against the annotated set during evaluation, where many good results were rejected. Similar problems appeared in personal names. Therefore we decided to perform another evaluation, with less strict matching criteria.

Table 2. Evaluation of extraction (intersect match)

Type	Total relevant	Total extracted	Relevant extracted	Recall [%]	Precision [%]	F1 [%]
Person	779	788	672	86.26	85.28	85.77
Telephone	262	178	178	67.94	100.00	80.91
Fax	139	127	125	89.93	98.43	93.98

In the second evaluation, we considered the information extraction result to be relevant when it intersected the manual annotation. The results of evaluation were much better (Table 2) and showed us that if we enhance the email information extraction, we can get better recall and precision, as well as better results in the social network extractor.

4.2. Social Network Extraction

Here the test task consisted of assigning telephone numbers to persons. For both the personal names and phones the social network extractor depended on the information extractor (IE). Its only responsibility was to correctly assign the received phone numbers to the received personal names. This difference in the task (compared to IE) necessitated a different evaluation methodology. Our evaluation of social network extractor focused on user needs and was inspired by the approach used to evaluate the TREC 2006 Enterprise Track in [8]. We defined the user needs as the ability to extract attributes (phone numbers) and assign them correctly to primary entities (persons). It is not necessary that all the occurrences of each attribute value or primary entity be identified in the email archive. It is acceptable that some occurrences are missed so long as each *unique* attribute value is identified and correctly assigned to primary entity. The telephone number was considered correctly assigned if any of the acceptable variants of its owner's name appeared as the first or second in the list of potential "owners" sorted by score. The results are summarized in the tables below.

Table 3. Quantitative evaluation of the extraction of telephone numbers for social network reconstruction

Total relevant	Total found	Relevant found	Recall [%]	Precision [%]
42	32	29	69	90.6

The number of telephone numbers in Table 3 differs from that in Table 2, because Table 3 represents only the *unique* telephone numbers for the whole set of 50 emails. Moreover, the telephone numbers were re-formatted by leaving only numeric characters in them, before they were added to the social graph.

Out of 42 relevant *unique* phone numbers in the test corpus (and 32 phone numbers actually extracted), two occurred in the emails that did not contain any personal name (only company name was present), so they were irrelevant for the task of pairing phones and personal names, which is summarized in Table 4.

Since there were 40 relevant *unique* phone numbers, we expected 40 correct pairs consisting of a phone number and a personal name. This was for us the "Total relevant" for the pairing task. The number of assigned pairs (shown in the "Total Found" column) is now 30, since it excludes the two irrelevant phone numbers. The number of correctly assigned phones to persons is shown in the "Relevant found" column.

Table 4. Quantitative evaluation of the assignment of phones to personal names

Total relevant	Total found	Relevant found	Recall [%]	Precision [%]
40	30	18	45	60

Table 4 implies that out of 30 found pairs, 12 were wrong. Out of these, 6 were caused by the fact that the Information Extractor failed to extract the name of the person that actually "owned" the phone number. Further 3 errors were caused by the extraction of wrong (corrupted) phone numbers, as implied by Table 3. This means that out of 21 pairs which the social network extractor could possibly pair correctly, it only failed 3 times. This would give the precision of 18/21 = 85.7%, which demonstrates the potential of spreading activation in reconstructing social networks. The immediate conclusion is that we need to focus primarily on improving the quality of the information extraction of text-based data in multilingual contexts.

4.3. Summary

In contrast to [7], where we tested our prototype on a set of 28 emails written in English, we now tested it on a set of 50 emails written in Spanish, supplied by our Commius project partner, Fedit. Spanish emails, with their accented characters and differing cultural norms represented a challenge. Some of them were solved on the fly, but others require a substantial redesign of the information extraction strategy. The 60% precision of the social network extraction is less than what we obtained with English emails in [7] (77% precision), but there are clear possibilities for improvement, especially at the information extraction stage, which is the base for next steps. The evaluation of the information extraction from emails is our primary focus, and we have noted several opportunities for the improvement of extraction patterns, which will then give much better results. We expect the improvement to be as good as the one shown in Table 2, since most of the errors in the social network extractor were due to the lower recall of the preceding information extraction step. Spreading activation can deal with the lower precision of the information extraction but cannot cope with low recall, i.e. the situations when something needed was not extracted. In this respect, the main opportunity for improvement lies in making the partially good results of the information extraction (summarized in Table 2) available for the social network extractor, since now only strict IE results (shown in Table 1) are really exploited for social network reconstruction.

5. Conclusion

In this paper we discussed email social network extraction based on information extraction and spread activation.

The "Social Network Extractor" component that we developed is able to process either mailboxes in mbox format or directories with email (.eml) messages, and thus extract multidimensional social network information contained in the email archive. In such a graph or network it is possible to see and exploit the links among objects such as people, time, email addresses, subjects, URLs, contact details or recipients.

The preliminary results of the extraction of social networks from email archives show that it is possible to deliver Xobni-like functionality in the enterprise or organizational context. Our approach is based on the concept of spreading activation similar to IBM Galaxy [5]. We have shown inferring relations between people and phone numbers on a limited set of emails using a simple algorithm. The success rate (precision) of the experiment was not as high as we have expected (only 60%) but on simpler English emails in [7] the precision was higher (77%). We believe there is a possibility for improvement by fine-tuning the information extraction and spreading activation algorithm. In future, we would like to infer the relations such as those between customers and services, suppliers, products and transactions, organizations and people, people and address details, and others. The approach can deliver the information needed for the adaptation of enterprise systems by filling in enterprise system databases upon installation and thus help them to offer full functionality from the beginning.

Acknowledgments

This work is partially supported by projects Commius FP7-213876, APVV DO7RP-0005-08, AIIA APVV-0216-07, VEGA 2/0184/10 and VEGA 2/0211/09. We would also like to thank Fedit for providing us with emails for testing.

10. References

[1] Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A., "Mining Email Social Networks", In: *MSR '06: Proceedings of the 2006 Workshop on Mining Software Repositories*. ACM, New York (2006) 137–143.

[2] Culotta, A., Bekkerman, R., McCallum, A.: "Extracting Social Networks and Contact Information from Email and the Web". In: *CEAS'04*. http://www.ceas.cc/papers-2004/176.pdf

[3] Laclavík, M., Šeleng, M., Ciglan, M., Hluchý, L., "Supporting Collaboration by Large Scale Email Analysis", In: *CGW'08* (2009) 382-387. ISBN 978-83-61433-00-2

[4] Diehl, C. P., Namata, G., Getoor, L., "Relationship Identification for Social Network Discovery" In: *The AAAI 2008 Workshop on Enhanced Messaging, AAAI Conference On Artificial Intelligence*, pp 546-552 (2008)

[5] Judge, J., Sogrin, M., Troussov, A.: "Galaxy: IBM Ontological Network Miner" In: *Proceedings of the 1st Conference on Social Semantic Web*, Volume P-113 of Lecture Notes in In-formatics (LNI) series (ISSN 16175468, ISBN 9783-88579207-9). (2007)

[6] Laclavik M., Seleng M., Ciglan M., Hluchy L.: "Ontea: Platform for Pattern based Automated Semantic Annotation"; In *Computing and Informatics*, Vol. 28, 2009, 555–579

[7] Kvassay, M., Laclavík, M., Dlugolinský, Š.: "Reconstructing Social Networks from Emails". In DATESO 2010: *Proceedings of the 10th annual workshop,* pp 50-59, (2010). ISBN 978-80-7378-116-3

[8] Soboroff, I., de Vries, A.P., Craswell, N.: "Overview of the TREC 2006 Enterprise Track". (URL: http:// trec.nist.gov/pubs/trec15/papers/ENT06.OVERVIEW.pdf)