Interpreting Contact Details out of E-mail Signature Blocks

Gaëlle Recourcé
Kwaga
15 rue JB Berlier
75013 PARIS
+33677259962

recource@kwaga.com

ABSTRACT

This paper describes a fully automated process of address book enrichment by means of information extraction in e-mail signature blocks. The main issues we tackle are signature block detection, named entites tagging, mapping with a specific person, standardizing the details and auto-updating of the address book. We adopted a symbolic approach for NLP modules. We describe how the process was designed to handle multiple-type of errors (human or computer-driven) while aiming at 100% precision rate. Last, we tackle the question of automatic updating confronted to users rights over their own data.

Categories and Subject Descriptors

[I.2.8] [Problem Solving, Control Methods, and Search]: Dynamic programming, Heuristic methods; [I.2.5] [Programming Languages and Software]: Expert system tools and techniques; [I.2.7] Natural Language Processing; [H.4.3] [Communications Applications]: Electronic mail; [H.1.2] User/Machine Systems: Human factors, Human information processing

General Terms

Application, Robustness, Dynamic Knowledge, Human Factors, Standardization, Expert System Reasonning.

Keywords

Contact details calculation; address book population; email signature ; sig block; named entities recognition; UIMA; NLP;

1. INTRODUCTION

In the framework of an e-mail processing chain, we implemented a cloud-based service for signature dectection in e-mail bodies and signature analysis (decomposition in typed named entities), vcard formatting and automatic merge into the user address book.

Aside from the e-mail importing and merge component, the process is encapsulated in an Unstructured Information Management Application allowing interchangeability of each component and scalability.

The process is fully implemented and running since May 2011. We visit approx. 500,000 e-mails a day. New features are regularly added to the original design but the overall architecture and purpose remain the same.

The service is available for user Gmail and Google Apps IMAP servers. Only French and English languages are fully covered but any ISO-latin signature can be analyzed (with potentially less quality, though, due to limitations in body pre-processing and linguistic analysis).

2. SPOTTING AND ANALYSING SIGNATURE BLOCKS

2.1 Context

Originally described in email text-to-speech context (see [3] and [4]), the signature block identification task is not as trivial as one might think, mainly because of the lack of formal definition: signature blocks are not explicitly described in MIME format[1] Contrary to From, To, Cc or Subject header fields, those information are freely added by the user (manually or via the software) in the mail body, in a particularly free way. Signature block were first described in Usenet conventions[2] along with strong recommendation to use the "dash-dash-space-carriage return" pattern, but were not formally encoded.

Moreover, signatures may appear at various places in the body: in new emails, they are auto-inserted at the end of the body, but may be followed by other insertions such as provider footer or corporate disclaimers. In reply emails, sig blocks are inserted either immediately after last user text insertion (that is before quoted text) or at the end of the body (just as in a new email).

The difficulty of the task is increased by the fact that signatures are not compulsory (there migth be no signature at all) and because there could be several signatures in one email.

The first signatures were only text formatted. Along with the evolution of email usage, HTML signatures are now quite common, as well as insertion of image (logo or picture)[3].

Last, long-line signature may be broken by spacing or character conversion realized by messenging systems, inducing complex rules to reconstruct original formatting.

The following table shows the number of detected vCards for 4 users with different incoming e-mail volume.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

[1] Cf. http://en.wikipedia.org/wiki/MIME - RFC 2045 through 2049.

[2] Cf. http://tools.ietf.org/html/rfc3676#section-4.3

[3] Our system do not cover image analysis nor the .vcf attachment technique. We only considered HTML or text signature inside the BODY.

Table 1. Percentage of vCards detected in incoming emails

User	Number of incoming emails	% with vCard	Number of non-bac'n e-mails	% with vCard
User A	45 518	22%	24 207	41 %
User B	16 090	6%	3 629	28%
User C	3 452	8%	813	33%
User D	17 033	6%	2 365	45%

2.2 Wide Coverage Context

Part of our motivation to choose a symbolic approach rests in the wide coverage our system: the set of e-mail we analyze may be written in English or French but also in other languages provided the body text is mainly iso-latin (few other characters may be included). Emails are often multi-lingual (either because it includes a translation or because signature in one language and message body in an other language) and we had to adapt the linguistic process to that observation: rules and dictionaries has to be extended to be robust and accurate in both languages.

The geographical origin of the e-mails is potentially worldwide, which lead us to implement local convention for phone number or postal address format. It's a work in progress: we only cover accurately approx. 20 countries but the necessary diversity of the corpora we have to cover lead us very strict error handling strategy.

2.3 Signature Analysis Process

Several tasks are performed on the e-mail before the signature analysis:

1. Extraction of HTML Part from MIME format (or text otherwise)
2. Elimination of specific configurations (encoding not supported)
3. Language detection. We use for this task the TextCat guesser[4] that we adapt to email by training on a set of emails.

The output is an xmi object (UIMA representation for documentation analysis annotation). Our process for signature analysis is a tree-fold UIMA annotator[5]:

First, we preprocess the e-mail body to detect new text, quoted text, signature zones and footer. We'll see below how this first calculation may be reconsidered the third phase if some inconsistencies were detected. This component is very similar to previous work in the e-mail segmenting task.

Linguistic analysis is implemented through an open-source FSA-based corpus processor Unitex[6] called through a UIMA annotator. The analyzis consist of:

1. Segmentation and tokenisation (specific to e-mail text)
2. Named entities recognition (names, roles, organization, postal and email addresses, fax or phone number, identifier on social networks, urls (blog, site, profile, …))
3. Calculation of a set of linguistic hints allowing the determination of a category for e-mail content.

Third annotator is the interpretation of those data in expert system style. This module is described in the next section.

The duration rate for UIMA analysis over a sample of 7743 e-mails is 299ms/mail.

3. INTERPRETATION

3.1 Presentation

The main steps in the interpretation component are:

1. Corrections on the preprocessing calculation (new text vs quotes)
2. Grabbing the signature, i.e. spotting the beginning and end of the sig block and modelizing each named entity (phone, address, name) inside this zone. Some text may be non-relevant for contact creation such as quotations, tweets, corporate advertisement, …).
3. Cleaning and normalizing contact details.
4. Deciding if contact is valid (coherent and well attributed) and formatting it in vCard.

We have to deal with erroneous matter in the interpretation phase. Following [5], we adopted dynamic knowledge model correction: instead of waiting perfect data or doing backtracking when it's not the case, we chose to set a dynamic model that adapts the knowledge in specific conditions.

Our main guideline was not to add any false information in the user address book. We, therefore, had sometimes to prefer silence over noisy update.

The following sections highlight some aspects of the interpretaion component.

3.2 Whose Details Are Those?

The accurate association of the set of details found in signature with a specific person or contact in the address book was the first issue we confronted when implementing the auto-updating address book process.

The task is complex mainly because of the combination of two factors: email addresses are frequently missing in signature blocks and an address book contact must have (is identified by) an email address.

We thus established as a first rule that the sig block details were to be attached to the 'From' header field. This raises three kinds of difficulties :

1. The first one is related to the processing chain itself: there are some inevitable errors in the body analysis which lead to errors in signature spotting: those signatures may be from a previous sender in the same thread. We had to implement specific rules to prevent incoherent mapping.

[4] http://odur.let.rug.nl/~vannoord/TextCat/

[5] Currently incubating at Apache Foundation, UIMA, http://uima.apache.org/index.html, was originally designed by IBM http://domino.research.ibm.com/comm/research_projects.nsf/pages/uima.index.html)

[6] Unitex is a corpus processing system, based on automata-oriented technology, conceived and implemented by a team of computer scientists from LIGM at Univ. Marne-la-Vallée and distributed under LGPL license. (full list of contributors)

2. Some patterns were found in emails that refers to specific situations: address change, contact transfer, out-of-office notifications, leaving company announcement… In all those emails, contacts details are not straightforwardly related to the email sender.
3. ‘From’-header field can be misleading: for instance, automatic notifications e-mails often come with real names but artificial e-mail address (such as “*John Doe <gramlab@alerts.assembla.com>*”). Using this kind of information would lead to spoil the address book with useless contacts.

3.3 Formatting Details in vCard

3.3.1 Vcard Format

Once the coherence of contacts details actually checked, data from are formatted into vcard 3.0 format[7]. Fields are standard or customized.

Table 2. vCard 3.0 format contact fields

Field	Content	Standard/ Custom
N, FN, NAME	Name (with internal structure), Formatted name, Name	Standard
TEL tel-type	Phone number	Standard
EMAIL	E-mail address	Standard
ADR	Postal address	Standard
TITLE	Job title, functional position or function	Standard
ROLE	Occupation	Standard
ORG	Organizational name and units	Standard
X-URL	Specialized by domain (e.g. Facebook) or type (e.g. Blog)	Custom
X-ID	Specialized by provider (e.g. SKYPE-ID, TWITTER-ID)	Custom

We also standardize each field in the vCard. It means for instance, that postal addresses are structured with internal fields (street, town, postal code, …). This allows to enrich address book with proper presentation (strongly dependant on the geographical location). This standardization is also a necessary step for smart comparison with existing information in the address book.

As an example, we present in section 3.5 the normalization and internationalization for phone numbers.

3.4 Standardizing Phone Numbers.

Two motives lead us to standardize phone numbers when updating user address book.

First, to ease automated processes: the auto-dial function, for instance, won’t work well with human number formatting. Second, to facilitate human international communication by adding international prefix when missing.

We designed phone number standardization as a three-step process:

1. Detection of country or countries. Note that multiple-country signature are quite common, this use case cannot be disregarded.
2. Cleaning the phone number.
3. Adding the international prefix onto cleaned number if one and only country was found (by use of Google’s library libphonenumber)[8].

4. UPDATING THE ADDRESS BOOK

Last task in the process, address book updating presents the same ambivalent difficulty caused by conceptual complexity combined with idiosyncratic human usage.

First issue is to determine whether it is a new entry or an update of an existing one.

Most address book automatically create a new entry for each e-mail address detected in e-mail either out or in. Most users consider that two contacts with different e-mail addresses but concerning the same person (same full name usually) as being duplicates. Some address book software (or independent services) offer de-duplicate features, but it has to be activated manually (thus depending on user choice) and is not part of the automated collecting process.

As a result, most address books contain a mix of automatic (e-mail address based) and manual (name-based) classification.

Theoretically, the name-based technique raises the homonymie issue (two persons may have the same name but not the same address), because human full name are not unique contrary to email addresses. Empirically, this phenomenon is quite rare compared to the number of people having more than one e-mail address.

The updating process raises one more question: for each field in the vcard, one has to decide whether it is new information (new field) or an update (modify an existing field).

If it is a unique-type field (e.g. name, org or role):

- fill in if empty
- modify with normalized form if automatically generated
- don’t modify if it is user created

If it is a multiple-value kind of information (e.g. phone or addresses)

- add a new field if different
- normalize if identical or shorter

We use fuzzy match operation for phone number and comparison algorithm specific for postal address.

Empirical difficulties are numerous in this phase, both technical and philosophical:

- Existing duplicates: as said above, contact software naturally create duplicates, the fusion being human driven.
- Human peculiarity (e.g. adding notes inside phone number field)
- Geographical diversity (e.g. international phone number cannot be use for local call in Canada)

[7] http://tools.ietf.org/html/draft-ietf-vcarddav-vcardrev-06

[8] http://code.google.com/p/libphonenumber/

- Shared contacts and corporate directories: specific rights are associated to some contact groups in corporate gmail, such as corporate directory contacts that are visible but not mofiable by users.
- Along with the integration of Google+ contact in Gmail, Google started to modify some formerly user-private fields such as name with the Google+ profile information.
- Synchronization with other devices is a very important issue for all mobile users. Unfortunately, this process tends to give birth to messed up contacts, creating unpredictable configuration in the user address book.

In this process, we had to find the right balance between the absolute respect for user choices and necessary correction for mistakes.

5. CONCLUSION AND PERSPECTIVES

Our first objective while implementing this service was not to damage any of the user content. This drove most of our implementing decision.

Noisy information is not acceptable for the user: precision should always be preferred even if it implies some sort of silence (no detection or no update). In other terms, if a piece of information is not 100% sure at any level, we do not use it.

Robustness is our second priority: millions of emails were analyzed by our servers, specific rules were adopted: non-isolatin encoding or above-200kO emails, for instance, are not analyzed for the sake of robustness.

6. ACKNOWLEDGMENTS

This paper describes a team work implemented in 2010-11 by Kwaga's developpers, Philippe Laval, Frédéric Meunier, Sylvain Surcin, Fanny Grandry and Mohamed Hadrouj. Its interdisciplinary position results from the team multiple scientific fields: natural language processing, artificial intelligence, and computer science combined with a general concern with user needs.

7. REFERENCES

[1] Paumier, S., Nakamura, T., and Voyatzi, St.. *UNITEX, a Corpus Processing System with Multi-Lingual Linguistic Resources*. eLexicography in the 21st century: new challenges, new applications (eLEX'09). 22-24 October 2009. 173–175.

[2] Carvalho, V. and Cohen, W. 2004, *Learning to Extract Signature and Reply Lines from Email*, in Proc. CEAS, 2004.

[3] Chen, H., Hu, J., and Sproat, R. W. 1998. *E-mail signature block analysis. In ICPR'98*. Vol. 2, 1153-1156.

[4] Chen, H., Hu, J., and Sproat, R. W. 1999. *Integrating Geometrical and Linguistic Analysis for E-Mail* Signature Block. In *ACM Transactions on Information Systems (TOIS).* Vol. 17.4, 343-366.

[5] Pierret-golbreich, Chr. 1996. *Correction of Dynamic Knowledge.* In *Eight International Conference on Tools with Artificial Intelligence*. 228-233.