Extracting Information for Context-aware Meeting Preparation

Simon Scerri[1], Behrang Q. Zadeh[2], Maciej Dabrowski[3], Ismael Rivera[3]

[1]Fraunhofer IAIS,
Scholls Birlinghoven,
Sankt Augustin, Germany.
scerri@iai.uni-bonn.de

[2] INSIGHT Centre for Data Analytics,
National University of Ireland Galway,
IDA Business Park, Galway, Ireland.
{firstname.lastname}@deri.org

[3]Altocloud,
NUI Galway, Ireland.
macdab@altocloud.com
iriviera@altocloud.com

Abstract

People working in an office environment suffer from large volumes of information that they need to manage and access. Frequently, the problem is due to machines not being able to recognise the many implicit relationships between office artefacts, and also due to them not being aware of the context surrounding them. In order to expose these relationships and enrich artefact context, text analytics can be employed over semi-structured and unstructured content, including free text. In this paper, we explain how this strategy is applied and partly evaluated for a specific use-case: supporting the attendees of a calendar event to prepare for the meeting.

Keywords: Text Analytics, Context Awareness, Ontologies

1. Introduction

The objective of most context-aware systems is to utilise acquired context information in order to provide context-aware support to people (Hong et al., 2009). In this paper, we show how personal information typically available within a virtual office environment can be integrated, processed and enriched with context information, in order to support people with their day-to-day tasks. In particular, we focus on a specific use-case: preparing for a meeting that is scheduled[1] to start within a pre-defined time-frame. Frequently, people prepare for these events by trying to retrieve relevant material, and recall previous discussions and commitments. To automatically assist with this task, we require as much context information surrounding the scheduled event as possible.

Aside from the event's date and time, most calendaring tools are integrated with email services, and are able to link event participants to the existing contact lists. Additionally, the provided event title and/or description can be analysed to determine the event's *topic*. In particular, a set of terms extracted for an event can be compared to terms extracted from other personal text-based office items, e.g. documents, folder names, email and instant messages (IM) – especially those that are linked to the other participants (e.g. emails exchanged with the event organiser). The most relevant items can thus be retrieved and presented to the user before the event takes place. However, presenting a ranked list of related items can still be daunting. Therefore, in addition to displaying the co-occurring terms/people for each retrieved item, a form of item summarization is employed to highlight embedded implicit knowledge. In particular, we extract action items from retrieved email/IM messages – as these can indicate past commitments and expectations related to the event.

The above strategy requires two forms of Information Extraction (IE):

i. The first IE task targets structured/semi-structured information from legacy data, services and applications (e.g. email/chat messages/events/folders/documents and their attributes).
ii. The second IE task targets complex/unstructured information embedded within personal item titles, descriptions and content (e.g., keywords and action items).

In this paper we report on the latter effort, with a description of the two implemented information extraction services, and the results of their evaluation.

2. Background: System Architecture

To provide the envisaged support, the heterogenic nature of personal information in a person's virtual office environment had to be addressed. Most of this information is derived from native OS file systems, applications, online services and communication protocols. Specifically, a *Personal Information Item Extractor* (PIIE) targets text-based documents from the file system and calendar and email items from supported services (iCal, IMAP). In addition, the PIIE also intercepts Message Stanzas exchanged over an Extensible Messaging and Presence Protocol (XMPP) Server[2] to extract and process IM messages. Fig. 1. shows the general architecture of the implemented system. The personal information item extraction stage is marked by 'A'. The PIIE extends the Nepomuk project[3] architecture, which employed the Aperture Framework[4] in order to extract personal information, albeit limited to a personal computer (Sintek et al., 2009). Adopting the Resource

[1] We target users who rely on the use of office calendar application/service to manage, schedule and share events.

[2] http://xmpp.org/xmpp-software/servers/

[3] http://nepomuk.kde.org/

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

Description Framework [5] (RDF) as a standard representation format enables all retrieved personal information items to be integrated and processed centrally.

A number of existing/extended ontologies have been integrated to model all the required domains. Accordingly, the PIIE performs the routine semantic 'lifting' of new/modified personal information items onto a machine-processable representation, which is stored in a centralised RDF store. The PIIE employs two separate services (which is marked 'B' in Fig.1) to extract further implicit attributes from unstructured text that is associated with each information item. The Action Item Extraction (AIE) service operates on extracted email and IM messages; the Term Extraction (TE) service operates on documents (titles, content), email (subject, content) and IM messages, as well as calendar entries (title, description). Subsequently, the semantically enriched personal information item descriptions are then transferred to the Central RDF Store for later processing (marked 'C' in Fig. 1).

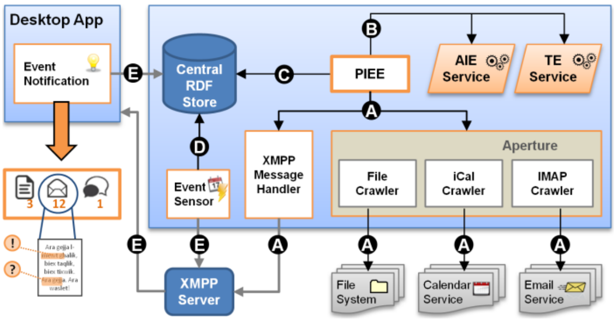


Figure 1: System Architecture

Following the extraction processes, an *Event Sensor* continuously checks ('D' in Fig. 1) for upcoming events that start within a predefined timeframe. When such an event is detected ('E'), an XMPP stanza containing the calendar event identifier is sent to the user's chat client over the XMPP server. An embedded event notification component will then retrieve the required information from the RDF store, including documents, emails and chats related to the event, through appropriate SPARQL queries[6].

Related items are retrieved based on:

a) term similarity, and
b) involved persons.

The former corresponds to term co-occurrence between the title/description of the upcoming event and the filename/subject/content of documents, email and messages – as extracted by the TE service.

The other criterion for retrieving related items consists of overlaps between the event participants/creators and the sender/recipient/participant of extracted email and IM messages. Subsequently, the retrieved items are ranked by relevance, according to the overall amount of term and person co-occurrences. For each relevant email and IM chat retrieved, we also highlight the embedded action items extracted by the AIE service – these are useful to remind the reader of the most pressing previously-communicated information available on the event.

In this paper, we explain the information extraction methodology used for the semantic enrichment of relevant personal information items, i.e. email and IMs, documents and calendar events.

3. Action Item Extraction

The AIE service operates on *Email* and *Instant Messages*. In this section we focus on the techniques behind the AIE and report on its evaluation.

3.1 Technique

The AIE service adapts an earlier component (Scerri et al., 2010) targeting the extraction of action items from email messages to i) address this task's more generic requirements and ii) also extend it towards instant messages. The existing component is driven by a classification model that maps extracted text clauses to one of various action items defined in (Scerri et al., 2008). The model behind these action items was designed in view of the requirements to support the workflows behind entire email threads. Thus, each action item carries explicit semantics corresponding to specific expectations, possibly requiring follow-up, for the message sender, recipient or both. In contrast, our objective is merely to visualise embedded social requests and expectations in relation to an occurring event, with no need for follow-up support or interaction. In this context, the AIE reduces the original action item set to the following three:

1. ***Information Request***: a text clause representing a request for information by the message sender.
2. ***Task Request***: a text clause represents a request for a resource to be sent, an event to be planned or held, a task to be planned or undertaken – either using the recipient informationor jointly by both sender and recipient.
3. ***Information Delivery***: an informative text clause that is not classified as either of the above.

The original action item extraction service described in (Scerri et al., 2010) is implemented as a rule-based action item classifier using finite state transducers in the GATE framework (Cunningham et al., 2002). It is based on a declarative model that considers sentence modality, pre-defined categories of action verbs (corresponding to separate gazetteer lists), tense, negation and semantic role. It consists of an ANNIE Corpus IE Pipeline which includes the standard GATE English tokeniser, sentence splitter, Hepple POS tagger and named entity transducer; in addition to the ANNIE gazetteer lookup and a set of

[5] http://www.w3.org/RDF/
[6] http://www.w3.org/TR/rdf-sparql-query/

hand-coded JAPE (Cunningham, 1999) grammars. A gazetteer list was constructed to bind special kinds of tokens/gazetteer entries to intermediate annotations (e.g. groups various types of verbs, modal verbs, and grammatical persons by category).

The implemented JAPE grammars match combinations of the linguistic/semantic annotations output by the above pipeline components to classify clauses into one email action item. The grammars, which number 58 rules in 14 phases also provide basic support for ‘if-then’ clauses in order to change the classification accordingly. An additional 2 phases, consisting of 4 rules, were introduced in the extended AIE service in order to map the original action items into one of the above-listed three. The gazetteer entries were also extended to cover terms written in so-called ‘text speak’ form, which tends to be more widespread in IM messages.

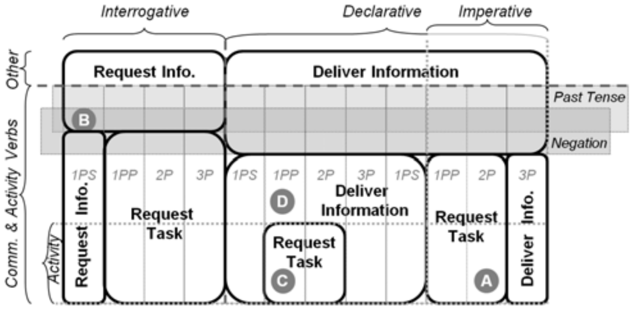


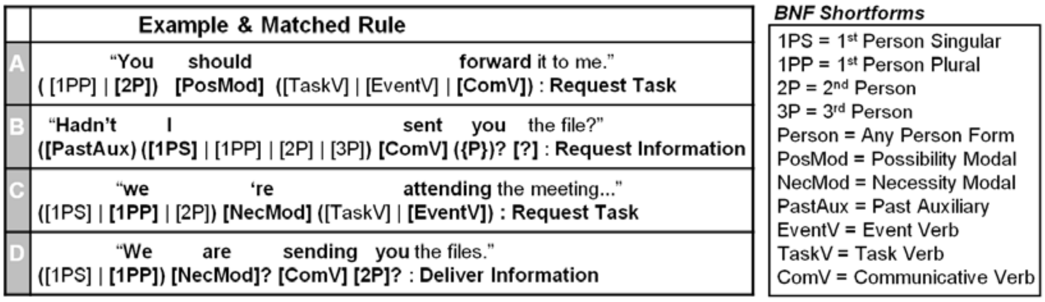
Figure 2: Visualisation of the AIE textual clause classifier, mapping text to one of three action items

The rule-based action item classifier, as customised for this task, is visualised in Fig 2. It breaks down the linguistic space into a number of dimensions, such that clauses are classified into one of the three action items. Vertically, the linguistic space is split by sentence modality: interrogatives, declaratives, and imperatives (as a subset of the latter). Horizontally, the space is divided between ‘Communicative’-type verbs (e.g. ‘send’, ‘forward’), Activity-type verbs (e.g., ‘attend’, ‘prepare’), as enlisted in two pre-defined gazetteer entries; and their complement (shown as ‘Other’). Statements having activity and communicative verbs are further segmented (vertically) based on the semantic role of the implicated agent, i.e., who is expected to perform the action. We differ between the following number and person: 1st person singular (1PS), plural (1PP), 2nd (2P) and 3rd person (3P). The effect of negation and grammatical past tense on a clause is demonstrated by the overlapping horizontal shades of grey across the figure. In the AIE, the behaviour of clauses containing communicative and activity verbs is almost identical, with the exception of non-imperative, declarative clauses implying the 1st person plural or a 2nd person as an agent of an activity. Examples of this different behaviour are shown in Table 1 (C and D). All shown examples (A-D) are linked to the corresponding spaces marked in Fig.2. The matching rule is shown below each example in BNF notation.

3.2 Evaluation

Repeating the evaluation process for the classifier in (Scerri et al., 2010), we employed 8 people to review automatic annotations generated for their personal email and IM chats, by rating the results and marking action items that they deemed missing. The AIE service used for the evaluation is available online[7]. As expected, given the reduced amount of action items, the resulting F_1 measure (0.64) is significantly higher than that for the original email classifier (0.58) reported in (Scerri et al., 2010). Furthermore, in the described use-case the extracted action items are only employed to generate visual reminders, and not to prompt any interaction or trigger automated tasks. Given this reduced risk, we favour recall over precision and therefore are more interested in the F_2 measure, which at 0.69, is well within the acceptable range amongst IE tasks of a similar complexity (Cunningham, 2005).

Another expected result is the higher score achieved when limiting the input only to email content (0.71), in comparison to processing only IM content (0.64), due to its less formal, less structured nature. This results suggests that extending the gazetteer with IM text short forms is not sufficient, and that the difference in between email-style statements and their IM equivalent is more than superficial. Thus, the JAPE grammars behind the AIE need to be re-examined to improve the accuracy of action item recognition in this form of communication.



	Example & Matched Rule
A	“**You** **should** **forward** it to me.” ([1PP] | **[2P]**) **[PosMod]** ([TaskV] | [EventV] | **[ComV]**) : **Request Task**
B	“**Hadn’t** **I** **sent** **you** the file?” (**[PastAux]**) (**[1PS]** | [1PP] | [2P] | [3P]) **[ComV]** (**{P}**)? **[?]** : **Request Information**
C	“**we** **‘re** **attending** the meeting...” ([1PS] | **[1PP]** | [2P]) **[NecMod]** ([TaskV] | **[EventV]**) : **Request Task**
D	“**We** **are** **sending** **you** the files.” ([1PS] | **[1PP]**) **[NecMod]**? **[ComV]** **[2P]**? : **Deliver Information**

BNF Shortforms

1PS = 1st Person Singular
1PP = 1st Person Plural
2P = 2nd Person
3P = 3rd Person
Person = Any Person Form
PosMod = Possibility Modal
NecMod = Necessity Modal
PastAux = Past Auxiliary
EventV = Event Verb
TaskV = Task Verb
ComV = Communicative Verb

Table 1: BNF-notation of some rules, with examples of how textual clauses are classified.

[7] http://srvgal66.deri.ie:8080/SemantaEvaluation/

4. Term Extraction

The term extraction (TE) service operates on *Email* and *Document* content, as well as the descriptions of *Calendar Events*. This section describes the employed method for the implementation of the TE service.

4.1 Technique

The term extraction process consists of two steps. In the first step, candidate terms are chosen. In the second step, each candidate term is assessed, weighted and marked as valid or invalid term. In the first step, a linguistic filter is used to choose sequences of tokens with certain POS tags as candidate terms. In the second step, in order to assign weights to candidate terms and classify them as valid or invalid term, the TE service employs a distributional approach.

It is assumed that the context of terms that are known prior to the extraction task can be used to identify new key terms. Therefore, a vector space model is implemented to capture the co-occurrences of terms and words that define the context of (previously known) valid and invalid terms. Following the construction of the vector space model, for the assessment of terms' contextual similarity, a support vector machine (SVM) is employed to measure terms' contextual similarity.

In the implemented system, words surrounding a target term in a text window of size *n = 3* are considered to form the term's context. Text windows are expanded symmetrically in both sides of candidate terms. In addition to the words in the neighbourhood of candidate terms, the co-occurrences of words and terms that stand in a grammatical relation, regardless of their distances, is also included in the model as a term's context. As suggested by Maynard & Ananiadou (2000), in order to capture terminological knowledge around valid and invalid terms, the model is also enriched with co-occurrences count of terms and bigrams that are extracted from a text window of size 5 around valid terms. As a result, in the classification task, a high-dimensional sparse vector space – on the orders of hundreds of thousands in which most of the elements of the vectors are zero – represents target terms.

The major danger in such a scenario is over-fitting. The number of SVM model parameters to be learned increases as the dimension of the vector space increases. As a result, training an SVM using a high-dimensional sparse vector space results in a model that has poor predictive performance. In order to avoid over-fitting, the original higher-dimensional input vectors are required to be transformed into lower-dimensional vectors. We employ random indexing, a method of dimensionality reduction that is based on random projection (Sahlgren, 2005). Instead of the construction of vectors that represent terms at the original high dimension, they are directly constructed at the reduced dimension.

The construction of the vectors consists of two steps. In the first step, each of the employed contexts in the model, i.e. a co-occurring word with a term, is assigned uniquely to a randomly generated vector, which is called "*index vector*". Most of the elements of an index vectors are set to 0 except *1/s* elements that are set to equal number of 1 and -1, where s is the square root of the number of employed contexts in the model. In the second step, in order to represent a target term at reduced dimension, the term is assigned to an empty vector called "context vector". The context vector has the same dimension as index vectors have. When the term and a word/bigram co-occur, the context vector of the term is accumulated by the index vector of the word/bigram. The accumulation of all the index vectors of the words/bigrams that co-occurred with the target term results in a context vector that represents the term at reduced dimension.

In order to develop a SVM reference model, we first construct the context vectors of the previously annotated valid and invalid terms in the training corpus. Each occurrence of a term in the training corpus is represented by a context vector that is marked by the class label of the term, i.e. valid or invalid. The set of constructed vectors from the training set are then presented to a linear SVM with the L2-loss function. After the learning procedure, the resulted model is used for the classification of the new terms that appear in the text stream. We employ the LibSVM and its implemented class membership probability estimation (Chan & Lin, 2011*).*

The major cause of imprecision in the output of the TE service is the erroneous determination of candidate terms. When a corpus is available for processing, a sequence of tokens can be tested using a statistical measure such as Log Likelihood Ratio in order to verify their stability as a lexical unit – known as unithood measure in automat term recognition. At the absence of a corpus, specifically at the beginning of conversations, we employ the frequency of candidate terms in a given chat history or email conversation to assess their unithood.

In order to process an input text stream, candidate terms are first extracted. Each candidate term in a sentence, is represented by a context vector. The context vector is then classified using the SVM model and it is assigned a class membership weight w_{ci}. In order to alleviate the problem of unithood measurement, candidate terms are grouped by string matching. A candidate term of t token length and frequency of f is assigned to an overall weight $w_u = \sum_{i=1}^{f} \log(t+1)\, w_{ci}$. The top m terms from the list of candidate terms sorted in descending order by their weight w_u value are then chosen as the representative keywords for the text that is being analysed. We choose *m* according to the length of the input text and a threshold on the w_u.

5. Conclusions and Future Work

The two services described in this paper are employed by the PIIE of the latest Meeting Assistant prototype. One of the contributions of this paper is related to an initial evaluation of these services and applicability of chosen approaches for the targeted scenario. The evaluation and results reported here highlighted a number of shortcomings that will be addressed to improve both i) the items returned and ii) their internal annotations.

Due to a distinctive difference in the form of language used in email and IM communication (Quan-Haase, 2005), a secondary pipeline will be adapted for IM messages. The streaming of email and IM messages is bound to improve the overall accuracy of the action item annotation process. Specifically, we intend to employ a service[8] that performs abbreviation substitution based on a dictionary of internet slang and acronyms, both of which are associated with IM chats.

Term extraction will be improved through an iterative learning procedure; whereby the term extraction process will be indirectly validated through the desktop app. Terms extracted from items that are ignored/selected by the user after the described retrieval process, will be labelled as in/valid terms respectively, and added as records in the training set. The goal is to investigate possible improvements for term extraction at the presence of noise, which will be introduced by the suggested technique for training data set expansion. Lexical association metrics measures can also be applied to filter candidate terms, once enough data has been archived for a statistical interpretation.

6. Acknowledgements

This work is supported in part by the European Commission under the Seventh Framework Program FP7/2007-2013 (digital.me - ICT-257787) and in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Líon-2) and SFI/12/RC/2289.

7. References

Chang, C.C and Lin C.J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* (TIST), 2(3).

Cunningham, H. (1999). JAPE: a Java Annotation Patterns Engine. *Research Memorandum*, Department of Computer Science, University of Sheffield.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics*, Philadelphia.

Cunningham, H. (2005). Information Extraction, Automatic. *Encyclopaedia of Language and Linguistics*.

Hong, J., Suh, E., Kim, S. (2009). Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36(4), pp. 8509--8522.

Maynard, D., & Ananiadou, S. (2000). Identifying terms by their family and friends. *Proceedings of the 18th conference on Computational linguistics*, 1, pp. 530--536. Association for Computational Linguistics.

Sahlgren, M. (2005). An introduction to random indexing. *In Methods and Applications of Semantic Indexing Workshop at the TKE 2005,* TermNet News, 87 :1-9.

Scerri, S., Mencke, M., Davis, B., and Handschuh, S. (2008). Evaluating the Ontology powering sMail - a Conceptual Framework for Semantic Email. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*.

Scerri, S., Gossen, G., Davis, B., and Handschuh, S. (2010). Classifying action items for Semantic Email. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*.

Sintek, M., Handschuh, S., Scerri, S., van Elst, L. (2009). Technologies for the Social Semantic Desktop. *Reasoning Web*, volume 5689 of Lecture Notes in Computer Science, pp. 222--254. Springer.

Qasemizadeh, B., Buitelaar, P., Chen, T., Bordea, G. (2012). Semi-Supervised Technical Term Tagging With Minimal User Feedback. *Proceedings of the 8th International Conference on Language Resources and Evaluation.*

Quan-Haase, A., Cothrel, J., and Wellman, B. (2005). Instant messaging for collaboration: A case study of a high-tech firm. J. *Computer-Mediated Communication*, 10(4).

Zadeh, B.Q. and Handschuhe S. (2014). Evaluation of Technology Term Recognition with Random Indexing, In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC).*

[8] The service provided by noslang.com is one valid candidate