Context Aware Knowledge Zoning: Traceability and Business Emails

François Rauscher, Nada Matta, Hassan Atifi

Context Aware Knowledge Zoning: Traceability and business Emails

François Rauscher, Nada Matta, Hassan Atifi,

Institute ICD/Tech-CICO, University of Technology of Troyes,
12 rue Marie Curie, CS 42060, 10010 Troyes Cedex, France,

{francois.rauscher, nada.matta, hassan.atifi}@utt.fr

Abstract. Even if immaterial capital represents an increasingly important part of the value of our enterprises, it's not always possible to store, trace or capture knowledge and expertise, for instance in middle sized projects. Email it still widely used in professional projects especially among geographically distributed teams. In this paper we present a novel approach to detect zones inside business emails where elements of knowledge are likely to be found. We define an enhanced context taking into account not only the email content and metadata but also the competencies of the users and their roles. Also linguistic pragmatic analysis is added to usual NLP technics. After describing our model and method, we apply it to a real life corpus and evaluate the results based on machine learning experiments and filtering algorithm

Keywords: Knowledge Engineering, Knowledge Management, Project memory, Traceability, Professional e-mails, Pragmatics analysis.

1 Introduction

During the 20th century the information society taken over the industrial society, and with the raise of the internet has been replaced by the knowledge society. Today there is a renewed interest in the knowledge-based systems, and therefore a growing need to question the sources of information. This interest is justified by the fact that organizations are evolving in world that is complex, uncertain and subject to rapid changes. Especially for companies, knowledge creation and acquisition (and therefore individual and collective learning) are playing an increasingly decisive role. They became fundamental elements and guarantee competitiveness and long-term performance. Implicit and explicit knowledge [1].

Artificial intelligence (AI) is a field of research based both on information technology and on social sciences and humanities researches. The AI techniques are routinely used in industry and we start to implement this methods in knowledge management [2].

The approach proposed in this article follows a problem encountered in the course of our work in the field of knowledge engineering in traceability of industrial projects.

adfa, p. 1, 2011.

These works involved advising and assisting support teams several years after the end of a project to find and organize information for their work. We had to help teams find methods, understand the decision made in the past. Our resources were primarily textual because team members were not easily available (departures, transfer, retirement). The text documents have the advantage of being easy to access and they reflect a general knowledge of the area although the authors do not always make it explicit. If text is the source of knowledge, knowledge engineering may uses tools from NLP (similar to the construction of ontology from texts [3]). However some of the data that we had to deal with consisted of email corpus for projects. And in this case, as pointed in [4] "the semantics of language cannot be reduced to a simple interpretation of syntactic forms in a set-universe." We use the framework of corporate and project memories to take into account an enhanced context and examine how we can find traces of collaborative knowledge in emails. In this study, we opted for an approach based on pragmatic linguistic, discourse analysis, organizational aspects (roles and competencies) to extract parameters that can permit us to locate traces of knowledge.

2 Related Works

In the following subsections, we review briefly which kind of knowledge traces we were looking for in the emails and how it fits with corporate memories and previous results.

2.1 Project memory

Compare to corporate memory, project memory (PM) [5] is a restricted part of a much larger capitalization exercise of a whole range of diverse experiences within the company. A project memory is generally defined as a representation of the experience gained during the implementation of projects [6]. It describes the history of a project and the experience gained during the implementation of a project [7]. This memory must contain elements of experience from both the context of problem solving. Project memory aims at traceability and reuse in similar project. Project memory used as materials the project data, the products of the project, stakeholders (role, competence in the organization, exchanges and meetings, electronic communications, telephone) and related documents. One sometimes encounters a reverse scale effect as the corporate memory: it is sometimes difficult to establish a method, software and collection of interviews for a project involving fewer than a dozen people. Tools and procedures exist for such tasks, but the size of the teams makes the systematic collection expensive and difficult. (Especially when the project is finished)

2.2 Competencies Modeling

In the PM frame, our approach is to enlarge the context of our search with organizational and human elements, like the roles, skills, competencies of the project members. Our simple hypothesis is that collaborative knowledge is more likely to be created when

some of people exchanging messages have the necessary competencies to solve the current problems.

Competency definition depends highly of the discipline (sociology, psychology, management) as stated in [8; 9]. In the perspective of human resources, competencies are the measurable or observable knowledge, skills, abilities, and behaviors (KSAB) necessary to achieve job performance. One can distinguish between soft competencies (managerial and social interaction), hard competencies (functional and technical specific to a field [10]. In our model, we will focus on technical competencies and their relations to the task that must be accomplished for the project.

Another axis of analysis are the relationships of the project members. Roles in the organization are important to our study because they could help detect indirect requests. For instance, if a manager is writing to a developer “I would like (...)”, it is for us one sign of an implicit request. Our model will take the roles into account using official function (hierarchical) or business relations (client/contractors).

2.3 Emails and pragmatics

Email is a medium that appeared at the same time as the World Wide Web. It has changed little since but despite its announced death many times (in favor of social networking or instant messaging), it is always present and important in business life. Some studies have looked at the email from a purely linguistic point of view [11] or communicative with the CDMA (Computer-Mediated Discourse Analysis) [12]. Main work on email have focused on spam management, classification, or topics clustering. In spite of its asynchronous and remote nature, S. Herring stated that email could be considered as form of discourse and thus make possible the application of conversational analysis techniques.

In fact most of the time the techniques used around the email borrows from NLP (Natural Language Processing) even if the message size is sometimes poorly adapted to the constraints of statistical analysis. Note the works of [13] on user preferences extractions. However, some authors have turned to another approach by using another branches of linguistic. We can cite for instance the research of [14] using N-grams to classify “email speech acts”, such as "propose a meeting" or "commit to a task". Similarly we can find in [15] works on email another grid of analysis coming from linguistic pragmatics and psychology, the Verbal Response Mode (VRM) from [16].

Kalia et al. [17] analyzed parts of Enron corpus, using two types of speech acts from Searle [18]: directive and promissive in order to identify tasks and commitments

Linguistic pragmatic origins from Searle's work and Austin [19]. A speech act is an act that is carried by saying it. Speech acts are categorized according to their motivation (order, promise, request, etc...). Pragmatics also emphasizes a fundamental basis of the theory of creation of organizational knowledge: the close relationship between language and human action in term of intention and implication of the learner. Compared to a simple parser (NLP) or even semantics, considering the goal sought by the speaker in a statement facilitates its interpretation. Our study is focused on the request speech act (direct or indirect) because it is present in the statement of a problem during the

lifetime of a project. In our coding scheme, a grammatical utterance corresponds to only one speech act as in Table 1

Request Form	**Linguistic form**	**Examples**
Direct request	Imperative	Do x
	Performative	I am asking you to do x.
	Want or Need statements.	I need/want you to do x
	Obligation statements	You have to do x
Indirect request	Query questions about ability of the hearer to do X	Can you do x? Could you do x?
	Query questions about Willingness of the Hearer to do X	Would you like to do x?
	Statements about the willingness (desire) of the speaker	I would like if you can do X I would appreciate if you can do X

Table 1. Request Speech Act custom coding scheme

3 Problem Statement

In this study, we are interested in medium-sized companies with projects involving geographically distributed teams, or remote workers (increasingly common for instance in IT development projects).

These companies consulted us with identified in terms of traceability and knowledge reuse but with some restrictions on resources. The projects reached completion several years ago, there were the deliverables / products, project data management (planning, documentation, and specifications), the list of participants and their skills, and their overall electronic exchange with related documents. The question was whether, in a professional email corpus, with project data context (team members, organization, document, product output), we could locate "knowledge" in these emails if so what type he was to automate this process and especially how to reuse it.

As stated above, we chose first to focus on problem solving [20], because useful knowledge for the business are most likely to be used or created during this kinds of exchanges. Especially on the undefined type of problem or "wicked problems" [21, 22], where collaborative knowledge occurs naturally.

We have developed an analysis model that includes the project data grid (organization, people involved with their roles and skills, email exchanges, produces and associ-

ated document) which allowed us to highlight the occurrence of problem-solving exchange, and associated context.In a similar manner as email zoning in Lampert study [23], our approach is to identify the "areas of knowledge" For this purpose, we will extract the parameters from analysis model and test two different types of algorithm.

At first we will apply Machine Learning technique on the whole vector of parameters and then we will compare it to an incremental filtering method. Machine Learning will be a supervised Support Vector Machine (SVM) like in [24] even if the corpus we had for training and testing was not very large. Finally once problem solving knowledge areas are detected via an algorithm, we will compare the result with manual labelling by experts.

4 Context Aware Knowledge Zoning

Our method aims at finding elements of reusable knowledge, that we call “knowledge zones” inside large corpus of professional emails that were collected during the lifetime of a project.

4.1 Overview

Fig. 1. We used a two steps approach: first with the help of an analysis grid that was designed with experts (in knowledge engineering and pragmatics) we built a features vector at the sentence/message level, then we used this feature vector as an input to determine if the sentence/message belongs to a knowledge zone (KZ) using both algorithm (A schema of CAKZ system and structure is presented in Figure 1).

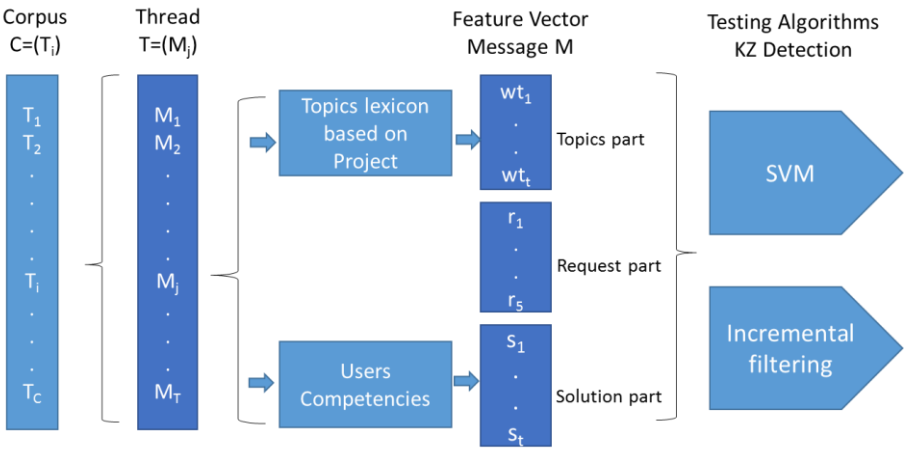

Fig. 1. Overview of CAKZ System

4.2 Common definition

An email corpus is a set of messages ordered by time of arrival and grouped by threads (i.e. initial message with replies).

In a thread T, consist of messages $(Mi)_{i\in T}$, each message M as an emitter E_M and receivers $R_M=(TO_M,CC_M)$ (respectively direct receivers (TO) and carbon copy (CC)). The BCC(blind carbon copy) are not used in this study.

4.3 Features vector

The features vectors have several parts: the "raw content" part: derived from the email content (e.g. body, subject, attachments names) that are projected into a weighted topics vector (based on our dictionary), the enhanced context that divide itself into subparts: request, roles, competencies.

Topics Features.
This part of the feature vector is made to answer the question: are the users dealing with an issue related to the topics of the projects? In order to capture that we first define the project topics. We will keep only threads containing more than three messages because we are using discourse analysis analogy and need to have a significant exchange occurring between the users. Then we will assert to deal with messages concerning pure software functionality knowledge and to filter out project coordination or irrelevant emails. Our approach is to create a keywords dictionary for the main topics (task and milestones) of the project. This dictionary can be built from the following sources:

- Project phasing and specifications documents
- an expert;
- domain ontology if available;

As in project memory context, we choose not to rely on statistical NLP clustering like in Cselle et al. [25] but to use existing context knowledge. This dictionary is voluntarily kept simple and have the form $D = (t_i)_{0\leq i<t}$, where t is the number of topics: Topics t_i : $keyword_{i1}$, $keyword_{i2}$... $keyword_{ip}$.

As a side remark, keywords chosen in topics shall not overlap too much to keep the results significant. Using this dictionary we classify messages into weighted topics vector. The content of all messages are represented by Vector Space Model [26], i.e. $M=(w_i)_{0\leq i<k}$, in which each term is weighted by its *tfidf* score, k being the size of the vocabulary.

The same is done for topics. We then compute a ranking between our messages and each topics. In order to do that we use a cosine similarity based algorithm. This give us a topics matrix T where (T_{ij}) represents weight of topic j in message i.

For the SVM, the vector T_{ij} will be used as topic input part for message i. For the filtering algorithm, the messages that will be kept for the next step must reach a threshold S1 at least in one topic, i.e. kept M_i where $\max_{1\leq j\leq t}(T_{ij}) \geq S1$

Requests Features.

As second step we are detecting requests signs. In order to do so, we establish with a linguist a set of custom features. Presence or absence of: interrogation sign, specific bigrams and trigrams based on pragmatic in Table 1 ("you should", "we must", "can you", etc.) and keywords ("question", "problem", "error" etc.). We also add a temporal part by taking into account if a request sign was already present in previous messages from the same thread. When users are facing a problem, they are usually asking a lot of questions before reaching an agreement on a solution.

Another custom feature to detect indirect request is the relation between emitters and receivers. A "relationship" matrix R by using a weighted directed graph representing both hierarchical and client/contractors links. Users are vertices of the graph and the weights on edges bring a measure of user/user "influence" (real values between 0 and 1). R_{ij} stands for the "ordering capacity" from user i to user j.

For the SVM, the 5 request features (interrogation sign, pragmatic sign, specific keywords, temporal, relation) will be used as request input part. For the filtering algorithm, for each thread T from previous step, we compute a score between 0 and 1 (combining and normalizing custom features) on each message. We kept threads containing messages with Prob(Request) ≥ threshold S2. This gave us a subset of threads TR having messages with a good probability of request. (potential boundary of a KZ)

Competencies Features.

In the last part of feature vector, in incremental filtering, we will look for pieces of knowledge related to the requests founds in the second step. For each thread T in TR, we have identify messages M_r where a request is likely to occur. We will then examine all the following messages in the same thread i.e. $M_{s,T} = (M_i)_{(i>r),T}$. taking into account user competencies.

First we built a matrix CU representing user competencies, (using curriculum vitae and function description of their role in the project). $CU = (CU_{ij})$ representing the skill level of user i in competence j. We took similar approach as Vergnaud [9] to measures skills (0=not knowing, 0.25 =novice, 0.5=medium, 0.75=experienced, 1= expert).

Then we built a matrix CT representing the competencies needed to fulfill a topic (its associated tasks) for the project, $CT = (CT_{ij})$ representing the importance of competency i regarding the topics j. This matrix is built with experts in each topics, again with discrete weighting (ranging from 0=competency useless for the tasks, 0.25, 0.5, 0.75, 1= vital competency). We then construct the matrix $UT = {}^{t}CU.CT$ where (UT_{ij}) stands for a very rough estimation of the skills of user i regarding the topic j.

For the SVM, the following vector $Vi=(UT_{ij}.T_{ij})$ representing emitter competency on topics vector will be computed on each message i. and account for the competency part of feature vector.

For the incremental filtering, we compute on each message $M_i \in M_{s,T}$, the following emitter score and compare it to a threshold S3: score $(E_{Mi}) = \max_{1\leq i\leq t}((UT_{ij}.T_{ij}) \geq S3$

We are dealing messages with potential solution of the problem solving raised by request in M_r, we are trying to assert that the emitter have the necessary competencies to bring new knowledge regarding the current topics.

This gave us a final subsets of messages M_k that are likely to contain traces of collaborative knowledge regarding the topics of the projects.

5 Experiment and Analysis

For the purpose of this study, we use a real world project in software development and apply our method on data. Data were collected after the end of the project that lasted nearly 2 years. A publishing group editing law-related codes (like insurance code, labor code) hired a software development company to create a workflow tool for their journalists. Due to geographical constraints, nearly all the communications during specification, implementation, tests and delivery were done through email.

5.1 Project Team

The team was split between the contractors and the development team. Among these, various roles and skills were present, but the main actors were:

- SRA: Chief editing manager (skill: law and management, Role: Contractor);
- CT: Law Journalist (skill: law and management, Role: Contractor end-user);
- JBJ: Information System Manager (Skill: Information system, Role: Contractor);
- BLA Information System Project Manager (Skill: Information system, Role: Contractor Employee);
- FX: Information System Developer (Skill: Software Engineering, Role: Development manager);

5.2 Corpus

Our corpus represent 3080 messages/ 14987 sentences in 801 threads between 30 projects actors.

Business emails collected from a project in their raw form are very redundant. In case of multiple replies or forward, several parts of the messages are repeated (e.g. quoted reply content). This occurs typically in long threads, mediated equivalents to spoken conversations, which are especially interesting for our study. Some preprocessing steps have to be performed in order to prepare messages and threads for analysis. We chose a deliberately simple method analog [Carvalho & Cohen, 2006]. The steps involved were:

- Remove all previous message text from reply;
- Keep previous message in case of first reply of a thread or forwarded email cause it carries context information;
- Remove signatures and disclaimers when possible (identity of sender and receivers are kept in email metadata);

This leaves us with a corpus of messages and threads without too much duplicated or useless information. For some treatments, the granularity at message level is not sufficient, and it's relevant to split the messages into sentences. Here again, we use a standard approach and split according to punctuation and paragraph signs

Topics and requests.
Based on project specifications and code deliveries, a small topics dictionary (as in Table 2) was built according to Section 4. 3 and the message topic T matrix was computed accordingly. Stemming was done to compute the term frequency and cosine similarity.

Topic	Keywords
XML	structuration, tag, tree, xsd, dtd, schema, markup, xml
BDD	Database, table, fields, editorial part, code part
Workflow	workflow, validation, collaboration
Code	Law, legifrance, insurance, chapter, article, annexes, labor code
Paper	Indesign, Xpress, print, template, styles, margin

Table 2. Excerpt from topics dictionary

For request, we built a feature vector as described in section 4.3 for each messages/sentence containing (interrogation mark, pragmatic verb marker, specific keywords, temporal part, max emitter influence over TO (direct) receivers).

For the filtering algorithm, to compute the final probability of request, interrogation mark was used first, then the remaining parameters.

Competencies.
Finally the two matrices CU and CT defined in section 4.4 were computed. This operation is done once and valid for the whole project.

	SRA	JBJ	FX	RTO	BL	CT	CV
XML/XSL	0	0	1	1	0	0	0
C#	0	0	1	0	0	0	0
SQL	0	0,25	1	0	0	0	0
Architecture	0	0	1	0	0	0	0
Law code	1	0	0	0	0	0,75	0,5
Law writing	1	0	0	0	0	1	0
Indesign	0,25	0	0,5	0	0,25	0	1
HTML	0	0	1	0,25	0,25	0	0

Table 3. Excerpt Competency User matrix.

	XML	BDD	Workflow	Code	Paper
XML/XSL	1	0	0	0	0,5
C#	0,25	0	1	0	1
SQL	0	1	0,5	0	0
Architecture	0,5	0,5	0,5	0	0
Law code	0	0,25	0,5	1	0
Law writing	0	0	0	1	0
Indesign	0	0	0	0	1
HTML	0	0	0,5	0	0

Table 4. Excerpt Competency Topics matrix.

On Tables 3 and 4, we can see for instance that the user SRA possess good competencies in law and some in InDesign, and that these competencies are important for the tasks in topics Code and Paper. The values of competencies here are arbitrary and only matter to compute a matching and a score with topics.

Corpus labelling.

We had to label the corpus for the supervised learning part and to evaluate performance. Analyzing performance in knowledge discovery in project memory is complicated. Pattern of knowledge traces useful for a given project are not well defined. For many real life tasks, manual annotation by an expert is the primary way of getting the labels, We kept a pragmatic and business oriented approach. At first we asked a linguist to manually label the corpus for the presence of direct/indirect request. That task was done at sentence level also.

Then we have to label the messages where answers to these requests were explored during the problem solving. In that case it is impossible to get the actual label (also known as the ground truth or gold standard) and it is estimated from the subjective opinion of a small number of experts. In order to do that, first we ask a technical expert to label messages that correspond to collaborative knowledge traces, then we find a manager that worked on this precise project 4 years ago to do the same.

Off course this approach is subjective to the annotator, but these answers are very project related and require different fields of expertise. The manual labelling is an expensive and time-consuming process. So we consider all the messages selected by both experts as good candidate for containing solutions to problems.

This part was especially tedious and due to time constraint only 70% of the corpus (starting from beginning in emails date order) was annotate by these last two experts.

Sample thread.

To better interpret the results, it is interesting to examine a sample thread on the filtering method. .In Table 5, elements of outputs from our algorithm are presented. The column R and C, stands respectively for Request probability and emitter Competency. We can see that a request is detected in the beginning of the conversation (then a second one) and so we look for possible answers in the next messages. In our experimental setting,

the topics are expressed as float vectors accounting for their weight in current message, but only the topics name are shown here.

From	To	Messages	Topics	R	C
SRA	FX	Linked text from mutuality code are inside !!!!!!without problem. little remark: : to close this case, could you give us the link texts from the automobile code? If not we will have to do it manually. S.	code	0,8	
cc:	JBJ				
JBJ	FX	i can't help here! FX, any idea?		0,6	
	SRA				
FX	SRA	we had the 26 texts from RTO (word) but only for mutuality code (..) You will find in attachement linked text to append to automobile code (..) you have to transform them into xml . Be very careful about the markups and the respect the xsd schema or the xsl stylesheets won't work (..) ps : annexes will be delivered overnight	xml, code		0,75
cc:	JBJ				
SRA	FX	(..) linked text from CTA : we will do it manually in automobile code. You can consider this question closed. S.	code		0,6
cc:	JBJ				

Table 5. sample thread output.

5.3 Results and performances

To measure the performances, we used standard precision (number of true positive results divided by the number of all positive results) and recall (true positive divided by the number of positive results that should have been returned) and compute F1 measure (harmonic mean of precision and recall) on the labelled part corpus.

The SVM was trained with 70% of the labelled corpus and tested on the remaining 30%. The dimension of the feature vector was 25. The results were not concluding at all: recall of 0.25. This is largely due to the fact that SVM was not the right fit for the task. First the size of the corpus was very small (but that is the size of real life project email corpus), secondly the corpus was unbalanced (number of KZ too low compare to the rest) and finally (and that is a business oriented reason), it is not possible in real office to have someone labelling part of the corpus before using it. And unfortunately the trained SVM cannot be used on another project.

As for the filtering algorithm, it performs slightly better. The first threshold S1 was set to 0.04 allowing a good ratio from signal to noise (a preliminary sampling was done on 100 randomly chosen messages). Nearly 15% of messages were discarded. The threshold S2 was set to 0.5. We identified 1426 potential requests. We took the messages following request and compute emitter score as explained in subsection 4.4. The last threshold S3 was delicate to tune because relying on number of users and topics. Again we randomly sampled 100 messages and set this threshold to 0.25. (validating with an expert the competencies of the emitters on given topics). If this threshold is increased it brings more messages with clues to solution of a problem but less likely to contain interesting information as the user may not have enough technical competencies to formulate it. On 2126 messages, precision =0.38, recall=0.55 and F1=0.45. These results are promising but far from being enough for a real world usage.

5.4 Discussion

With these preliminary results, we noticed we had large number of false positives. It came from two main causes: we detected too much requests (especially the indirect ones) and the fact we are doing message level analysis. Sometime in a response, the emitter have the sufficient competencies for the topics mentioned on the overall message, but the answers were not labelled as such by the experts because not relevant to current problem. For instance, the emitter can write a lot about xml and coding, but finally give information about a meeting or law code (where he/she's is not a specialist).. We also work on finding patterns in the request/answer/request/... within the same thread that are signs of problem solving event.

6 Conclusion and future works

In this article we proposed a model for analyzing professional emails from a project in order to locate problem-solving zones. Our study aims at traceability in the context of project memory. An analysis method was presented in order to choose significant pa-

rameters and we test this parameters with two types of algorithms (SVM and incremental filtering). In contrast to previous models of extracting meaningful elements (like tasks, or concepts, etc...) from emails by using pure NLP techniques, we emphasize on enhancing the context and incorporate pragmatics and organizational components (speech acts, competencies modelling, roles).

Numerical results shows that the task of detecting zone of knowledge is very delicate and subjective. However this task is necessary and is accomplished by employee in everyday life. This results further indicates that interesting pieces of knowledge doesn't always came from an initial request, and that requests and answers are often interweaved creating noisy context. Still there are some issues regarding the granularity (sentence or message level) of the approach requiring further study to achieve a better understanding of patterns of exchanges during collaborative knowledge creation. Another important factor is the temporal aspect, to decide if message is part of solution to a problem often required to take into account several messages above (or even previous threads).

As for the algorithmic part, we knew that SVM were not the perfect candidate but as they were used in many previous works, we thought it would an interesting comparison. However the small size of corpus inherent to projects and nature of email limit the potential for improvement in the supervised learning algorithm. Furthermore, this approach wouldn't fit in a real world scenario as no expert for labeling the train set would be available. The SVM might be used in future research as a subsystem part for request recognition on sentences.

While this study has shown relatively average results, we're planning further research studies to investigate the possibility of improving the algorithmic part. First by enlarging the context to surrounding messages and relevant threads and taking into account attachment's content. Second by making a more precise match between topics of request and possible answers. Finally by exploring unsupervised techniques from AI that would be more suitable for the task. It has become clear that taking only into account the textual content of email (and not the overall human context surrounding it) would lead to very limited results as detecting traces of collaborative knowledge. How emails could give new light of competencies, skills and organizational abilities of the members of project team remains an interesting topics for extensive study in the near future. The proposed system will also be evaluated on scenario where the project is yet not completed and will involve the users. Ultimately, our work contributes project memory traceability and structuration of knowledge in daily work realization of project.

REFERENCES

1. Nonaka, I., & Takeuchi, H.: La connaissance créatrice: la dynamique de l'entreprise apprenante. De Boeck Supérieur, (1997)
2. Liebowitz, J.: Knowledge management and its link to artificial intelligence. In Expert systems with applications, 20(1), 1-6. (2001)
3. Biebow, B., Szulman, S., & Clément, A. J.: TERMINAE: A linguistics-based tool for the building of a domain ontology. In Knowledge Acquisition, Modeling and Management (pp. 49-66). Springer Berlin Heidelberg, (1999)

4. Varet-Pietri, M.: L'ingénierie de la connaissance: la nouvelle épistémologie appliquée (Vol. 696). Presses Univ. Franche-Comté, (2000)
5. Matta, N., Ribiere, M., & Corby, O.: Définition d'un modèle de mémoire de projet, INRIA, RR-3720, (1999)
6. Dieng, R., Giboin, A., Amergé, C., Corby, O., Després, S., Alpay, L. & Lapalut, S.: Building of a corporate memory for traffic-accident analysis. In *AI magazine*, *19*(4), 81, (1998)
7. Pomian, F.: Mémoire d'entreprise, techniques et outils de la gestion du savoir. Sapientia, (1996)
8. Harzallah, M., Leclère, M., & Trichet, F.: CommOnCV: modelling the competencies underlying a curriculum vitae. In Proceedings of the 14th international conference on Software engineering and knowledge engineering (pp. 65-71), (2002)
9. Vergnaud, N., Harzallah, M., & Briand, H.: Modèle de gestion intégrée des compétences et connaissances. *EGC* (pp. 159-170), (2004)
10. Tripathi K., Agrawal M.: Competency Based Management In Organizational Context: A Literature Review. In Global Journal of Finance and Management. ISSN 0975-6477 Volume 6, Number 4 (2014), pp. 349-356. (2014)
11. Baron, N. S.: Letters by phone or speech by other means: The linguistics of email. In Language & Communication, 18(2), p133-170. (1998)
12. Herring, S. C., Barab, S., Kling, R., & Gray, J.: An Approach to Researching Online Behavior. Designing for virtual communities in the service of learning, 338. (2004)
13. Paik, W., Yilmazel, S., Brown, E., Poulin, M., Dubon, S., & Amice, C.: Applying natural language processing (nlp) based metadata extraction to automatically acquire user preferences. In Proceedings of the 1st international conference on Knowledge capture (pp. 116-122). ACM, (2001)
14. Carvalho, V. R., & Cohen, W.: Improving email speech acts analysis via n-gram selection. In Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech (pp. 35-41). Association for Computational Linguistics. (2006)
15. Felice, R. D., & Deane, P.: Identifying speech acts in emails: Toward automated scoring of the TOEIC® Email task. ETS Research Report Series, 2012(2), i-62, (2012)
16. Stiles, W. B.: Describing Talk: a taxonomy of verbal response modes. SAGE Series in Interpersonal Communication. SAGE Publications, (1992)
17. Kalia, A., Motahari Nezhad, H. R., Bartolini, C., Singh, M.: Identifying Business Tasks and Commitments from Email and Chat Conversations. In HP Labs Technical Report, (2013)
18. Searle, J. R. Speech acts: An essay in the philosophy of language (Vol. 626). Cambridge university press, (1969)
19. Austin, J. L.: How to do things with words (Vol. 367). Oxford university press,(1975)
20. Newell, A., Simon, H. A.: Human Problem Solving. New Jersey: Prentice-Hall, Inc,(1972)
21. Shum, S. B.: Representing hard-to-formalise, contextualised, multidisciplinary, organisational knowledge. In Proceedings of the AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, (1997).
22. Conklin, E. J., & Weil, W.: Wicked Problems: Naming the pain in organizations (White Paper): Group Decision Support Systems, (1997)
23. Lampert, A., Dale, R., & Paris, C.: Segmenting email message text into zones. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2 (pp. 919-928). Association for Computational Linguistics, (2009).
24. Lampert, A., Dale, R., & Paris, C.: Classifying speech acts using verbal response modes. In Australasian Language Technology Workshop, p. 34. (2006)

25. Cselle, G., Albrecht, K., Wattenhofer, R.: BuzzTrack: topic detection and tracking in email. In Proceedings of the 12th international conference on Intelligent user interfaces, p190-197. (2007)
26. Salton, G., & Buckley, C.: Term-weighting approaches in Automatic text retrieval. Information processing & management, 24(5), 513-523. (1998)