Formalized Conflicts Detection Based on the Analysis of Multiple Emails: An Approach Combining Statistics and Ontologies

Chahnez Zakaria[1], Olivier Curé[1], Gabriella Salzano[1], and Kamel Smaïli[2]

[1] Université Paris-Est, IGM Terre Digitale, Marne-la-Vallée, France
{chahnez.zakaria,olivier.cure,gabriella.salzano}@univ-mlv.fr
[2] Loria, Campus Scientifique, BP 239 54506 Vandoeuvre Lès-Nancy, France
smaili@loria.fr

Abstract. In Computer Supported Cooperative Work (CSCW), it is crucial for project leaders to detect conflicting situations as early as possible. Generally, this task is performed manually by studying a set of documents exchanged between team members. In this paper, we propose a full-fledged automatic solution that identifies documents, subjects and actors involved in relational conflicts. Our approach detects conflicts in emails, probably the most popular type of documents in CSCW, but the methods used can handle other text-based documents. These methods rely on the combination of statistical and ontological operations. The proposed solution is decomposed in several steps: (i) we enrich a simple negative emotion ontology with terms occuring in the corpus of emails, (ii) we categorize each conflicting email according to the concepts of this ontology and (iii) we identify emails, subjects and team members involved in conflicting emails using possibilistic description logic and a set of proposed measures. Each of these steps are evaluated and validated on concrete examples. Moreover, this approach's framework is generic and can be easily adapted to domains other than conflicts, e.g. security issues, and extended with operations making use of our proposed set of measures.

1 Introduction

Multinational enterprises have developed well since the emergence of globalization, i.e. the process by which local, regional or national phenomena become integrated on a global scale. In the early 90s, the total number of multinational entreprises exceeded 37.000 and they had more than 170.000 affiliates abroad [9]. This has led to the creation of virtual or geographically distributed teams that overcome the problems of distance by using Computer Supported Cooperative Work (CSCW) tools. However it is still difficult for a team leader to remotely manage the emotions of its members and the conflicts that may arise between them. Such situations can complicate communication and cooperation between them, and it affects their work efficiency.

R. Meersman, T. Dillon, P. Herrero (Eds.): OTM 2009, Part I, LNCS 5870, pp. 94–111, 2009.

During the experiments of Hawthorne [3], Elton Mayo studied the importance of emotions in the professional environment. This is opposed to the classical School of management, especially Taylor's model which created the symbol of the work dehumanization. Mayo proved that good horizontal and/or vertical relationships (i.e. between colleagues and/or between employ and his employer), in a professional environment, have a major influence on overall satisfaction provided by the work and personal productivity.

The constitution of virtual teams has accentuated the difficulty of understanding an employee's behaviour. Nevertheless, the team leader can overcome this situation with the data generated by the CSCW tools, especially through the analysis of emails which allow to generate important textual corpora due to its large exploitation in professional environments [23]. But the number of emails exchanged between team members on a daily basis can be so important that it may not be possible for a team leader to read them all. In fact we studied an e-collaborative work of educational content mediatization team[1]. Its members communicate using email and they put their leader in copy for all emails exchanged, allowing him to monitor their collaboration. This has generated a minimum of 40 emails daily.

Our solution, named Handling Conflict Email (HaCoEma), consists in the automatic detection of conflicts in emails exchanged between team members. Hence it enables team leaders to intervene and manage conflicts before they lead to irreversible situations. HaCoEma solves the task of conflict detection by classifying emails, according to a domain ontology of relational conflicts.

Using topic identification (TID) techniques may allow to identify conflict emails. The main objective of topic identification is to assign one or several topic labels to a set of textual data. Labels are chosen from a set of topics fixed a priori. Several approaches have been proposed at the end of 90's [26]. All these techniques use a metric which compares the document under processing with a list of topics. Our purpose here is to detect conflict in emails by considering a conflict as a specific topic which one has to identify.

In [2], we addressed the issue of email routing. It has been also considered as an identification problem and we showed the difficulty to process emails. Indeed, they have specific features which make them different from newspaper documents (which are in general the material raw for TID). In opposition to newspapers, it is not easy to find special email corpora. Obviously, everyone has a considerable list of emails in his mailbox, but for our purpose we need company's emails which are unfortunately not available for obvious confidentiality reasons. E-mails are often noisy which makes their interpretation uncertain. Hence, it is difficult to process them automatically in order to retrieve the most relevant information. Which information should be kept? Firstly, should all the headings which constitute the structure of an email be removed? Some of them could be very relevant for detecting the topic as subject, date, sender. Then, which likelihood can be attributed to them. In addition, emails are often ungrammatical, punctuation is usually missed, abbreviations are widely used, foreign words are utilized, images,

[1] http://ufc.dz/

web pages may be present, etc. All these problems make detecting conflicts in emails an interesting challenge to raise.

HaCoEma uses the TFIDF (Term Frequency-Inverse Document Frequency) principle and the SVM (Support Vector Machine) model to classify emails according to the concepts of an ontology of relational conflicts. Our study also addresses the issue of building an ontology of negative emotions, which is made up of two phases. First we conceptualize the domain by hand, then we enrich the ontology by using a trigger-based model which finds terms corresponding to different conflicts in corpora.

In many contexts, analysis of emails supports decision making, for instance to produce cooperative software, such as Bugzilla[2], an open bug tracking system, where the emails report bugs affecting Internet browsers, e.g. Mozilla, Firefox, Thunderbird. Therefore, we analyze the emails in the conceptual framework of Documents for Action (DofA) [27]. With this approach, we focus on the collective dimension of the writing process, to analyze emails in an asynchronous communication process between several agents sharing common interests.

The remainder of the paper is organized as follows. Section 2 presents concepts and technologies used in HaCoEma. Section 3 describes our conceptualization approach of the conflicts domain in two stages. Section 4 describes the models which we have developed and used for classifying emails based on the concepts of our ontology. It also validates the classification methods exploited with experimental results. Section 5 provides a solution to the analysis of multiple emails based on a set of proposed measures and possibilistic description logics. Section 6 discusses related work. Finally, Section 7 concludes with a discussion of future directions.

2 Background

2.1 Possibilistic Description Logic

In information technology, an ontology provides a shareable, reusable piece of knowledge about a specific domain and can be specified more or less formally in order to create an agreed-upon vocabulary. We have selected Description Logics (DL) [1] as a mean to represent ontologies in the context of this work. The choice of DL is motivated by the important number of available DL-based ontologies hence enabling cooperation between them and the increasing amount of associated tools, e.g. reasoners, editors, APIs.

DL corresponds to a family of knowledge representation formalisms allowing to present and reason over domain knowledge in a formally and well-understood way. Central DL notions are concepts (unary predicates), relationships, also called roles (binary predicates), and individuals. A standard DL knowledge base is usually defined as $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ (or TBox) and $\mathcal{A}$ (or ABox) consist respectively of a set of concept descriptions (resp. concept and role assertions).

[2] http://www.bugzilla.org/

The following concept description:

$$NegativeEmotion \sqsubseteq Thing \sqcap \forall hasForm.String \sqcap \forall hasSynonym.String \sqcap \forall hasCause.String$$

states that a concept named *NegativeEmotion* is described as the set of objects which have forms, causes and synonyms that are strings of characters. Concept assertions, for instance, are *Person(paul)* and *Email(email1)*; an example of a role assertion is *sentBy(email1,paul)*. Based on $\mathcal{K}$, some standard DL reasoning tasks are concept satisfiability, knowledge base consistency, concept subsumption and instance checking which are detailed in [1].

Possibilistic logic, or possibility theory, [8] provides an efficient solution for handling uncertain or prioritized formulas and coping with inconsistency. In this logic, each formula is associated to a real value in [0,1]. The notion of possibility distribution π, defined as $\pi : \Omega \rightarrow [0,1]$, where Ω represents the set of all classical interpretations, is fundamental in defining the logic's semantics. From this possibility distribution, two important measures can be computed: (i) the possibility degree of a formula ϕ, defined as $\Pi(\phi) = \max\{\pi(\omega) : \omega \models \phi\}$, where $\omega(\phi)$ is the degree of compatibility of interpretation ω with available beliefs. (ii) the certainty degree of a formula ϕ, defined as $N(\phi)$=1 - $\Pi(\neg\phi)$.

In possibilistic DL, a possibilistic formula is a pair (ϕ, α) where ϕ is a standard DL axiom, i.e. TBox or ABox axiom, and α expresses a degree of certainty. A set of possibilistic formulas, also called a possibilistic knowledge base ($\mathcal{PK}$), consists of a possibilistic TBox ($\mathcal{PT}$) and ABox ($\mathcal{PA}$). The classical knowledge base ($\mathcal{K}$) associated with ($\mathcal{PK}$) corresponds to $\{\phi_i | (\phi_i, \alpha_i) \in \mathcal{PK}\}$. A $\mathcal{PK}$ is consistent iff its $\mathcal{K}$ is consistent.

Given a $\mathcal{PK}$ and $\alpha \in [0,1]$, the α-cut of $\mathcal{PK}$, denoted $\mathcal{PK}_{\geq\alpha}$, is defined as $\mathcal{PK}_{\geq\alpha} = \{ \phi \in \mathcal{K} | (\phi, \beta) \in \mathcal{PK}$ and $\beta \geq \alpha\}$. The inconsistency degree of $\mathcal{PK}$, denoted $Inc(\mathcal{PK})$, is defined as $Inc(\mathcal{PK}) = \max\{\alpha_i : \mathcal{PK}_{\geq\alpha}$ is inconsistent$\}$.

Example: Consider a possibilistic DL knowledge base $\mathcal{PK} = \langle \mathcal{PT}, \mathcal{PA} \rangle$ where $\mathcal{PT}$= {(Email $\sqsubseteq$ ConflictEmail $\sqcup$ NonConflictEmail, 1), (ConflictEmail $\sqsubseteq$ $\neg$ NonConflictEmail,1)} and $\mathcal{PA}$ = {(ConflictEmail(email1), 0.7), NonConflictEmail(email1),0.3)}. The TBox $\mathcal{PT}$ states that it is certain that an email is either an email with or without conflicts and that conflict emails are disjoint from non conflict emails. The ABox $\mathcal{PA}$ states that the email identified by *email1* is more likely to be a conflict email (certainty of 0.7). Let $\alpha = 0.3$, we then have $\mathcal{PK}_{\geq 0.3} = \langle \mathcal{PT}_{\geq 0.3}, \mathcal{PA}_{\geq 0.3} \rangle$ where: $\mathcal{PT}_{\geq 0.3}$ ={Email $\sqsubseteq$ ConflictEmail $\sqcup$ NonConflictEmail, ConflictEmail $\sqsubseteq$ $\neg$ NonConflictEmail} and $\mathcal{PA}_{\geq 0.3}$ ={ConflictEmail(email1), NonConflictEmail(email1)}. It is clear that $\mathcal{PK}_{\geq 0.3}$ is inconsistent. Now let $\alpha = 0.7$. Then $\mathcal{PK}_{\geq 0.7} = \langle \mathcal{PT}_{\geq 0.7}, \mathcal{PA}_{\geq 0.7} \rangle$ where $\mathcal{PT}_{\geq 0.7}$ ={Email $\sqsubseteq$ ConflictEmail $\sqcup$ NonConflictEmail, ConflictEmail $\sqsubseteq$ $\neg$ NonConflictEmail} and $\mathcal{PA}_{\geq 0.7}$ ={ConflictEmail(email1)}. So $\mathcal{PK}_{\geq 0.7}$ is consistent. Moreover, $Inc(\mathcal{PK})$=0.3.

2.2 Statistical Models

Several approaches are proposed for building ontologies from corpora. They can be grouped into two categories: (i) structural approaches based on the use of formal grammar and (ii) non-structural approaches, such as statistical approaches which must use important enough corpora, in order to have reliable measures and find out interesting relationships between terms [13].

The acquisition of terms based on statistical approach exists since several decades: Enguehard and Pantera (1995) [10], Dias (2002) [7], etc. It consists on the idea that words of the same area tend to often occur together. Similarity measures are used to identify recurrent associations of terms. The correlated terms recurrences are extracted by using different kind of measures [21] like Mutual Information. It is a measure of distance stemming from the information theory, which allows to measure the degree of association between two events. The mutual information $MI(x, y)$ represents the importance of the relationship between two events x and y. The non-weighted MI is given below:

$$MI(x, y) = \log \frac{P(x, y)}{P(x)P(y)}$$

where $P(x)$ (resp. y) is the marginal probability of x (resp. y) and $P(x, y)$ is the joint probability of x and y.

In general, a classification model consists of two tasks: modeling the document using a model of representation, as the vector model [20], and his assignment to the topic that concerns through a classifier like SVM [4], or a distance measure like Salton's cosine [19].

The essential idea of SVM is to use kernel functions (such as the polynomial, the Gaussian, etc.) to transform not linearly separable data into linearly separable ones using a representation of higher dimension spaces. So the goal is to find a function F, to learn from observation of input-output and to predict other events. The function attempts to minimize the errors of learning while maximizing the margin separating the categories of data [4].

The representation model describes emails or documents with the terms of the vocabulary or the terms involved in the classification's topics. There are several weight functions that represent the importance of each term in the email, e.g. the occurrence frequency of the term in the email or the TFIDF measure. TFIDF is used to evaluate how important a term is to an email in a corpus. The importance increases proportionally to the number of times a term appears in the email but is offset by the frequency of the term in the corpus. It is calculated as follows [18]:

$$TFIDF(w_i, M) = TF(w_i, M) \times IDF(w_i) \quad where \quad IDF(w_i) = \log \frac{T}{t_i}$$

where $TF(w_i, M)$ is the frequency of the term w_i in the email M. T is the size of the corpus and t_i is the number of emails in which the term w_i occurs.

2.3 Evaluation Measure of Retrieval Systems

The combination of recall, precision and F-measure [13] is a popular evaluation for information retrieval systems. Recall is defined as the fraction of relevant emails that are retrieved by the system, precision is defined as the fraction of retrieved emails that are in fact relevant and F-measure characterizes the combined performance of recall and precision. These measures are calculated as follows:

$$Recall = \frac{Number\ of\ relevant\ emails\ retrieved}{Number\ of\ emails\ to\ retrieve}$$

$$Precision = \frac{Number\ of\ relevant\ emails\ retrieved}{Number\ of\ emails\ retrieved}$$

$$F\text{–}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

There are two other measures that estimate the performance of a system from its errors, namely the False Acceptance (FA), where an email is wrongly considered as conflictual, and the False Rejection (FR), where an email is wrongly rejected. These measures are calculated as follows [13]:

$$FA = \frac{Number\ of\ False\ Acceptances}{Number\ of\ emails\ retrieved} \quad FR = \frac{Number\ of\ False\ Rejections}{Number\ of\ emails\ to\ retrieve}$$

3 Ontology Construction

3.1 Manual Creation of the Skeleton of the Ontology

To the best of our knowledge, no ontology describes the domain of conflicts. Since such an ontology is required in HaCoEma, we had to design one. We achieved this task by focusing on the litterature on emotions and considering that conflicts are generally associated to negative emotions. To conceptualize the conflict domain, we based our work on the taxonomy of Michelle Larivey [14], but we changed the separation criteria of emotions. We can see in Figure 1 that a first criterion separates emotions according to the degree of conflict, the first category represents emotions that can produce substantial conflicts as *disgust* and *hatred*, the second one leads to anticipate some indirect conflicts as *indifference*. A second criterion separates personal from social emotions, in fact it distinguishes social emotions from other emotions. This is due to the fact that it is very difficult to determine a personal emotion. For instance, the *sadness* emotion may be social when this feeling is due to the behaviour of another colleague or friend, and may be personal when the person did not succeed to reach an objective; however, *jealousy* can easily be classified as social emotions.

Conceptualizing the field of conflict and the classification of emails were made in the French language. Figure 1 presents a translation into the English language of an excerpt of our ontology. In the next section we present the statistical model that we used to enrich our ontology from corpora.

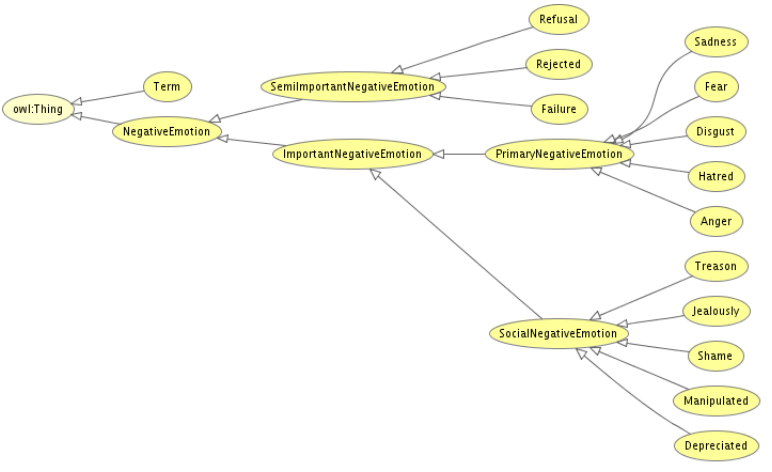

Fig. 1. Conflict ontology

3.2 Automatic Enrichment of the Ontology

Development of statistical language models is historically related to the construction of the first significant linguistic corpora [6]. For these models, a corpus represents a raw material, it is used to learn a maximum of linguistic events (n-grams, part of speech, etc.) [13]. In other words statistical processing of corpora allows to get knowledge by studying recurrent phenomenon. A corpus should be large in order to model statistically a maximum of reliable constructions. The more a corpus is important, the better the events are modeled [13]. For machine translation or speech recognition, it is not surprising to train the language model on a corpus with 300 million words. Classical n-grams models are often enriched by language models based on triggers which are used in several domains, e.g. in translation, and are exploited to build multilingual dictionaries [15].

We use a trigger approach [13] to enrich the ontology. Our aim is to find terms that are semantically related to the terms of the ontology, then to integrate them into the ontology, for a better description of its concepts. The triggers focus on terms that often appear together. That means we can predict the term w_j when w_i occurs (it can be written as: $w_i \rightarrow w_j$). For instance the term *"insult"* will probably predict the term *"humiliation"*. The triggers are determined by calculating for each ontology term its Mutual Information with each term in the dictionary. Then, only terms with a high Mutual Information are kept and used as triggered terms. We use this principle of trigger to enrich the ontology at the level of emotion concepts, because emotion is represented by just a few words which are synonyms, e.g. *"sadness"* emotion. Triggers also allow to collect several non synonym terms. Therefore we create properties to link them to the ontology and hence improve the classification of emails. Each triggered term will be manually associated to concepts of the ontology via *synonym* (regrouping some synonyms of the term representing the concept), *form* (regrouping terms

that indicate the expression of emotion) or *cause* (regrouping the reasons which may justify the expression of emotion) properties. For instance, the trigger model enables to enrich *Sadness* concept with the terms *grief, sorrow, etc.* as synonyms, the terms *suffer, endure, etc.* as forms and the terms *annoy, offend, etc.* as causes.

4 Classification of Emails

HaCoEma solves the task of detecting conflicts in emails by classifying them. It consists in identifying the concepts to which an email belongs to and therefore to recognize the emotion expressed in this email. The domain of classification is made up of two distinct approaches: supervised and unsupervised learning. The distinction between these two approaches comes from the knowledge or not of categories. Indeed, supervised classification learns to assign instances to predefined categories, while unsupervised classification is a task, which learns classification from the data, because categories are unknown. For the purposes of this paper we will focus on supervised learning. We classify emails according to concepts of the ontology, i.e. the categories of the classification are emotions of ontology.

4.1 TFIDF Classifier

Each email (E_i) to classify is encoded by a vector according to the terms of a concept (C_i). Then a similarity is calculated to quantify the semantic proximity between the email (its representation by the concept vector) and an emotion. This process is repeated for each emotion. Once all similarities are calculated, the classification process associates to each email the emotion with the highest similarity value. We introduce the following notations: $C_i = \{c_{i1}..., c_{ij}, ..., c_{in}\}$, where c_{ij} is the weight of the term w_j in the *ith* concept, and n is the number of terms in the concept which varies from one concept to another. $E_i = \{e_{i1}, ..., e_{ij}, ..., e_{in}\}$, where e_{ij} is the weight of the term w_j in the *ith* concept. Weights are estimated using the TFIDF. The classification is done by calculating for each pair (C_i, E_i) the cosine of the angle between vectors C_i and E_i defined as follows [19]:

$$Cos(C_i, E_i) = \frac{\sum_{j=1}^{n} c_{ij} e_{ij}}{\sqrt{\sum_{j=1}^{n} {c_{ij}}^2 \sum_{j=1}^{n} {e_{ij}}^2}}$$

4.2 SVM Classifier

The SVM are a class of algorithms inspired by the theory of statistical learning of Vapnik [4], it is a recent alternative for the classification and has been used in many applications such as face recognition, and bioinformatic. The SVM were originally designed as a binary classifier, however, they were generalized (SVMs) [22] [5] for a multi-class learning. For classifier with SVM, we used the Thorsten Joachims tool[3]. We represented our corpus in the input format of this tool, as follows:

[3] http://svmlight.joachims.org/svm_multiclass.html

```
<line>.=.<target> <feature>:<value>...<feature>:<value>
<target>.=.<integer>
<feature>.=.<integer>
<value>.=.<float>
```

The target value indicates the category of the email one of the classification categories, for example, the tuple: (3 1:0.43 3:0.12 9284:0.2), specifies an email of category 3 for which feature (term) number 1 has the value 0.43, feature (term) number 3 has the value 0.12 and feature (term) number 9284 has the value 0.2.

4.3 Evaluation

As explained before, it is difficult to get a corpus of emotions; in addition only those, which are subject to create conflicts between people, interest us. In order to evaluate our approach, we create an emotion corpus by extracting it from forum discussions. In fact, people in forums can exchange very hard words. To achieve that, we use our ontology as an index, which permits to collect all exchanges containing words predisposed to provoke a quarrelling between people. We then get 2.138 messages split into eight different emotions. In average each category contains 267 messages with a standard deviance of 16. Table 1 present the results of classification by using TFIDF and SVM. The results are presented in terms of recall, precision, F-measure, False acceptance and false reject. The first conclusion is that TFIDF outperforms SVM on our corpus. TFIDF achieves a F-measure of 0.93 whereas SVM gets 0.86. This could be very surprising in comparison to other works. This is due to the size of the corpus and the nature of the messages which are different from what we find classically in other works, e.g. processing of natural language. In fact, texts in our corpus are polluted making the frequency of words very low. It seems that TFIDF is less sensible to low frequency than SVM. Note that TFIDF allows getting a F-measure of 1 for *Hatred* concept and SVM achieves better results than TFIDF for *Anger* category.

Table 1. Performance of the TFIDF and SVM classifiers

Concept	**Recall**		**Precision**		**F-Measure**		**FA**		**FR**	
	TFIDF	SVM	TFIDF	SVM	TFIDF	SVM	TFIDF	SVM	TFIDF	SVM
Anger	0.92	1.0	0.96	0.93	0.94	0.96	0.04	0.07	0.08	0.0
Hatred	1.0	0.84	1.0	0.75	1.0	0.79	0.0	0.25	0.0	0.16
Treason	0.92	0.96	1.0	0.96	0.96	0.96	0.0	0.04	0.07	0.04
Jealousy	0.85	0.88	1.0	0.92	0.92	0.9	0.0	0.08	0.14	0.12
Fear	0.97	0.92	0.82	0.72	0.89	0.81	0.17	0.28	0.03	0.08
Depreciated	0.92	0.88	1.0	0.92	0.96	0.9	0.0	0.08	0.08	0.12
Sadness	0.92	0.52	0.92	0.87	0.92	0.65	0.08	0.13	0.08	0.48
Shame	0.96	0.88	0.89	0.88	0.92	0.88	0.11	0.12	0.04	0.12
Average on all the concepts	0.93	0.86	0.94	0.87	0.93	0.86	0.01	0.02	0.01	0.03

5 Analysis in the Context of Multiple Emails

In this section, we present a model for conflict management based on the analysis and classification of multiple emails, a set of measures and a possibilistic DL approach dealing with uncertainty.

5.1 Model Description

The *Email* class is the core of the UML model of Figure 2. An email is characterized by a date of writing, a subject and a body, which is composed by one or more fragments, having a suitable granularity. An email can be an answer to a previous email, or a forwarded email. In these cases, only the newly provided fragments of the body will be analyzed. Similarly, we do not analyze the attached file(s).

Many relations link the *Email* and *Agent* classes. In our application area, agents represent employees preparing the educational supports and the team leader. An email has one and only one sender and one or more direct receivers. An email can have one or more copy-receivers, who may be visible or not.

Depending on the presence of conflict terms, we classify the emails in two classes: *ConflictEmail* and *NonConflictEmail*. A conflict email is an email such that its body contains fragments related to some negative emotions, specified in the conflict ontology (Section 3). As the conflict emails are used to warn a conflict, *ConflictEmail* and *Conflict* classes are related. In practice, we also associate to a conflict email the emails having the same subject or subjects derived from it, such as Re(⟨subject⟩), Fwd(⟨subject⟩), Fwd(Re⟨subject⟩), and so on.

5.2 Required Measures to Warn a Conflict

The model in Figure 2 enables to warn about a potential conflict. In fact, we can compute some indicators, like as in the Table 2.

The indicators 1.3 and 1.4 give a global idea of the "width" and the "depth" of a conflict email, while the indicators 4.3 to 4.6 inform about the "width", "length", "density" and "urgency" of a conflict. These indicators enable the team manager to give a value to the *status* attribute of a conflict (current or completed), and to the *exit* value (satisfactory or unsatisfactory). We evaluate the "email agent activity" with indicators 5.6 to 5.8. A high conflict activity for an agent can imply his high implication in preventing conflicts (see for example the team manager, or an agent often involved because of his expertise). Moreover, an agent role in a conflict can be deduced from the relation (sentBy, sentTo, copyTo, blindCopyTo in the UML model of Figure 2) associating emails and agents. The indicators of Table 2 can be used to filter emails (respectively subjects, agents and conflicts), verifying some conditions based on the analysis and classification of the emails. Table 3 lists some examples of such conditions, where the values of α, β, γ and δ are chosen by the team leader.

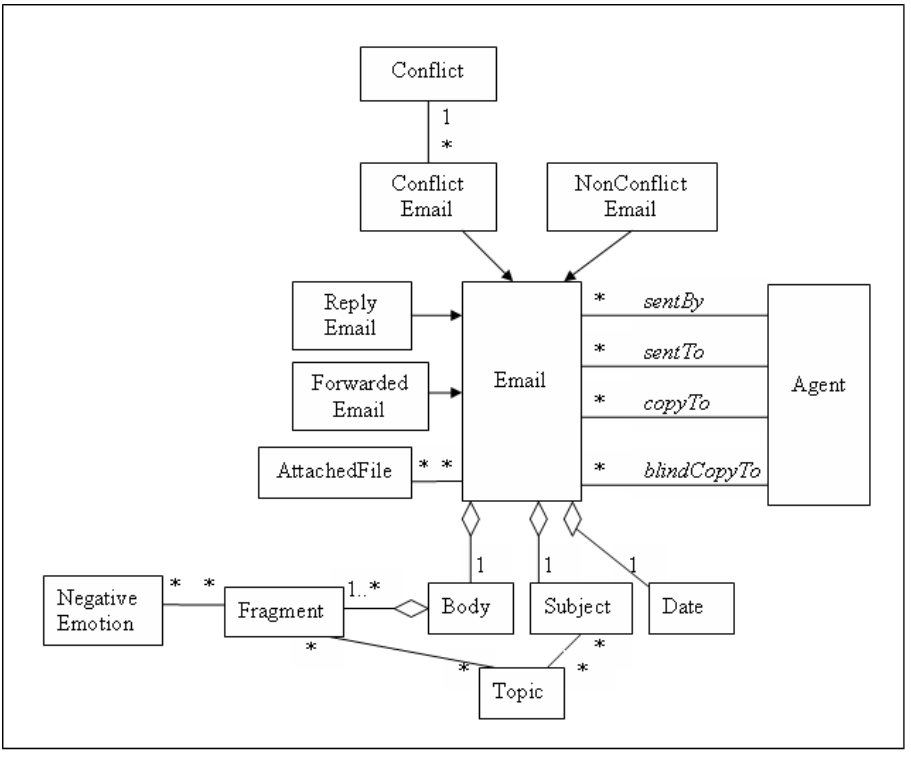

Fig. 2. UML model for conflict management

5.3 Ontology-Based Architecture

Since we are exploiting an ontology to represent conflicts, it seems relevant to develop an ontology-based approach to detect and warn conflicts between agents. This approach has several assets: (i) interoperability since every representation of knowledge exploits a DL formalism, (ii) representation of knowledge related to the fragment of social conflicts can benefit from the exploitation of ontologies developed in social networks, (iii) availability of efficient and user-friendly tools to design, maintain and process ontologies using standards of the Semantic Web.

Given Figure 2, we have everything related to conflicts and their classification represented in our ontology. The agent aspect of this model is already represented in the FOAF[4] RDF vocabulary standard. Moreover, this ontology can be used to represent relationships between (groups of) agents as well as information concerning projects they are working on. Hence, HaCoEma creates an ontology which contains some concepts relevant in our domain, e.g. *Email* and *DofA*, and relating them to the imported FOAF and Conflict ontology, e.g. $Email \sqsubseteq DofA$

[4] http://www.foaf-project.org/

Table 2. Indicator based on the analysis and the classification of the emails (TN stands for Total Number)

1. Email	1.1 TN of conflict terms 1.2 TN of fragments 1.3 TN of agents that receive email (directly or not) 1.4 TN Depth of conflict within an email = (TN of conflict terms) / (TN of fragments)
2. Date	2.1 TN of conflict emails at this date
3. Subject	3.1 TN of conflict emails with this subject (or a subject derived by applying Re and / or FWD) 3.2 TN of emails with this subject or a similar subject 3.3 TN of conflict emails with this subject / TN of emails with this subject
4. Conflict	4.1 TN of conflict emails related to it 4.2 TN of all emails related to it 4.3 TN of agents involved in this conflict =(TN of distinct agents who are senders of conflict email related to this conflict) + (TN of distinct agents who are (direct or not) receivers of a conflict email related to this conflict) 4.4 TN of days for this conflict = (Date of the last conflict email in this conflict) - (Date of the first conflict email in this conflict) 4.5 Density number (for a conflict) = (TN of conflict emails related to it) / (TN of all emails related to it)
5. Agent	5.1 TN of sent emails 5.2 TN of sent conflict emails 5.3 TN of received emails 5.4 TN of received conflict emails with a “send to” 5.5. TN of received conflict emails with a “blind copy to” 5.6 TN of emails in which sehe(he is involved = (TN of sent emails) + (TN of received emails) 5.7 TN of conflict emails in which she/he is involved = (TN of conflict emails that she/he sent) + (TN of conflict emails that she/he received) 5.8 Conflict activity =(TN of conflict emails in which she/he is involved) / (TN of emails in which she/he is involved)

and $DofA \sqsubseteq foaf : Document$. Finally datatype properties have been created to store values associated to the measures presented in Table 2.

Associated to this ontology, i.e. TBox, we can now create an ABox which stores all information according to a given state of emails exchanged between agents. It is also important to stress that we enable a team leader to enrich the ABox manually, i.e. he can assert that an agent or subject is conflicting and attach to this belief a certainty degree. Then, a set of emails, together with the agents sending and receiving them, can be envisioned as a graph.

Table 3. Examples of reports based on the analysis of the conflicts

Report	Conditions
Email	α_1 * total number of agents that receive it (directly of not) (1.3) + β_1 * Depth of conflict within an email (1.4) $\geq \delta_1$
Subject	α_2 * total number of conflict emails with the same subject (3.1) $\geq \delta_2$
Agent	Conflict activity (5.8) $\geq \delta_3$
Conflict	α_4 * total number of agents involved in this conflict (4.3) + β_4 * density number (4.5) + γ_4 * urgency number (4.6) $\geq \delta_4$

5.4 Use Cases Exploiting the Measures

We now present three use cases exploiting the measures proposed in Table 2. They enable to identify (i) agents which are involved in conflicts, (ii) subjects associated to conflicts and (iii) emails containing conflicts of certain types. These approaches exploit a subset of our proposed measures, features of possibilistic DL and its graph-based representation. We present the approaches exploited in the three use cases through an example whose's scenario is the following.

Scenario. A project leader supervises several persons which are exchanging emails on a daily basis. An extract of the information concerning these emails is represented in the graph of Figure 3: 3 persons (*paul , mary* and *peter*) and 3 emails are displayed. Each email has a subject property relating to the subject of the email and 3 datatype properties (one for each social, primary and semi importance type of emotions) storing a value in [0,1] and representing the certainty of appearance of such a conflict.

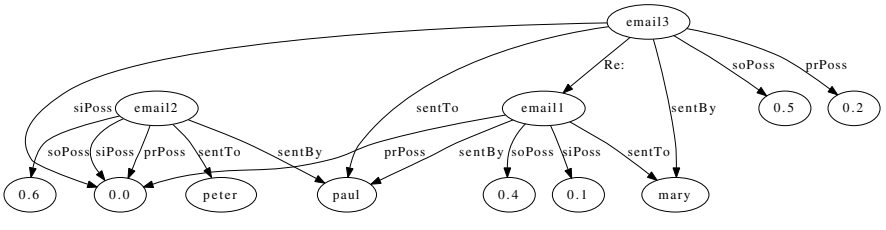

Fig. 3. Extract of an email graph

All use cases can be applied to predefined time periods. That is the emails' date have to satisfy some conditions, e.g. searching for conflicting emails between march 1st, 2009 and may 1st, 2009. This approach enables team leaders to search for agents, subjects and emails with conflicts over a given period of time.

Use Case 1: Agent Identification. This use case requires to compute the Conflict Activity (CA), i.e. 5.8 in Table 2. This is easily performed using a query language adapted to graph navigation, e.g. SPARQL's graph pattern

matching. Then for each agent, we assert a possibility formula stating that an agent is an instance of the concept *ConflictingAgent* with a certainty degree corresponding to his CA value. For instance, suppose that in our scenario, the agent denoted *paul* is involved in 5 emails and 3 of them have conflicts. Then the CA of *paul* is 0.6 and we can assert that: ConflictAgent(*paul*,0.6). Our ABox ($\mathcal{PA}$) also contains {ConflictAgent(*peter*, 0.2),(ConflictAgent(*mary*, 0.3), Agent(*paul*,1), Agent(*peter*,1), Agent(*mary*,1)} and an extract of the TBox ($\mathcal{PT}$) consists of {(ConflictAgent $\sqsubseteq$ Agent, 1)}. Then the inconsistency degree of $\mathcal{PK}$ is $Inc(\mathcal{PK}) = 0$ and all 3 agents are conflicting to a certain degree. Now suppose that the team leader has defined that *mary* is certainly not an instance of *ConflictAgent* $(\neg ConflictAgent(mary), 0.8)$. Then $Inc(\mathcal{PK}) = 0.3$ and only *paul* can be considered to be conflicting.

Additionally to this possibility theory approach, we also exploit some user defined preferences corresponding to the values α, β, γ and δ of Table 5. In the context of agent identification, δ_3 is a relevant parameter that enables to restrict the set of agents identified as conflicting. For instance in our running example, if $\delta_3 \geq 0.7$ then no agents are displayed to the team leader.

Use Case 2: Subject Identification. In this use case, for each subject with at least 2 emails, the system: (i) computes the measure 3.3 in Table 2, denoted ψ, by navigating into the graph. (ii) creates a possibilistic instance with type *ConflictSubject* for this subject. (iii) sets ψ as the certainty degree of this concept. Identifying conflicting subjects is then similar to the approach presented for agent identification.

Use Case 3: Email Identification. In this use case, the team leader selects the type of conflict he is interested in, e.g. social, primary or semi-important. Then the possibility value associated to this type, respectively *soPoss,prPoss* and *siPoss* in the email graph, are used to defined *ConflictEmail* possibilitic assertions for each email. The operations used to identify emails is then similar to the ones presented in the agent and subject approaches.

Summary. We have presented 3 use cases based on measures related to the emergence of conflicts. We believe that they correspond to the fundamental metrics expected from team leaders willing to minimize and prevent conflict situation in their teams. The architecture adopted for these 3 use cases is quite general and can easily be reused and extended to define finer grained solutions based on the measures of Table 2.

6 Related Work

In this section we provide a succinct overview over related works. In the first part, we analyze HaCoEma with respect to a system visualizing discussion content and participant behavior, while in the second part we interpret emails in the conceptual framework of Document for Action. In a third part we present different types of work on the building of emotion ontologies for classification.

6.1 Communication Garden

Systems that improve Computed Mediated Communication (CMC) should work on the archives of emails on different aspects (discussion content, participant behavior, social network between participants) and at different layers (information representation, classification or visualization). Generally, only some aspects and layers are treated by the existing systems. With respect to this framework, our research focalizes on the first two layers (conflict email representation and classification), and it enables the analysis of participant behaviors in a social network. The treatment of email content is deferred to our future works (Section 7).

To evaluate a computer mediated organization, the "Communication Garden System" [28] visualizes discussion content and participant behavior. It makes use of the very suggestive "flower metaphor", to represent graphically the liveliness of the threads of communication and the persons' activities. A thread is represented as a flower, where the number of petals equals the number of messages posted for this thread and the number of leaves represents the number of persons participating to the discussion. The starting time and the topic area are displayed by the localization of the thread. Similarly, the "person flower" represents some statistics (number of messages, number of discussions, duration). The visualization system allows suggesting easily "the current hot topic" in the community or the "most active" participants, by using statistics similar to the indicators shown in Table 2.

6.2 Documents for Action

In accordance with [27], emails can be considered as documents used to mediate the coordination of a "community of actions", i.e. a widely distributed group committed to working towards a common goal. Manuel Zacklad distinguishes two kinds of goals, services goals and integration goals : the first tend to be reached as the result of epistemic transactions, while the second rely on relational transactions. Operational and strategic activities are associated to the services goals, while relational and integrative activities are associated to the integration goals. In our application area, we associate the service goals and the operational activities (making educational supports) to the contributor agents. Through the email activity (relational activity), these agents and team leader participate of the integration goals. The team leader plays a special role, through his integrative activities, because he regulates the organization by analyzing conflict emails and solving conflict situations. In this context, emails participate to collective activities, transmit and disseminate information quickly, help in decision making and demonstrate situations. Then an email has all the properties of a Document for Action (DofA) : (a) sustainability, due to the participants commitment ; (b) fragmentation, due to the body structure, Replay and Forwarded emails ; (c) non trivial relations between the email, producers and receivers ; (d) extended state of incompleteness (chain of emails). In concordance with the conceptual framework of DofA, the email model (Figure 2) determine some dependencies between emails and agents, as listed on Table 4. An automatic visualization of these dependencies can be a useful support for the team leader.

Table 4. Examples of dependencies between emails and agents

Type of dependencies	If	Then
Email/Email	e_1 replies (or forwards) to e_2	e_1 depends on e_2
Email/Agent	e_1 is sent by a_1	e_1 depends on a_1
Agent/Email	e_1 is sent to (or copied to) a_2	a_2 depends on e_1
Agent/Agent	e_1 is sent by a_1 and e_1 is sent to (or copied to) a_2	a_2 depends on a_2

6.3 Emotions for Classification

Until recently researchers have ignored the emotional message behind the communication. However, the understanding and expression of emotion is not only important for humans, but is also critical for human-computer interaction. It is studied in different areas, such as psychology, neurology and sociology. Although several description models of emotions exist, categorical [16] [25] and dimensional [11] [12] models are the most commonly encountered. Most of the research on emotion analysis has been done on the applications of machine learning to emotion classification. In [24], affective lexicon ontology is constructed to classify emotional texts with SVM, it includes 10.200 entries and it is used to analyse the text from three different levels : words, sentences and discourses. In [25], Chinese emotion ontology is used in classifying the emotion of the actors in sentences. It is built from HowNet, and it contains just under 5.500 verb concepts covering 113 different emotion categories. [11] proposed an emotional ontology, where each emotional concept is defined in terms of a range of values along three emotional dimensions, corresponding to evaluation, activation and power. Classification and ontology are used to provide particular rules from emotional text. These rules provide configuration parameters for a system for emotional voice synthesis. The classification is used in the other types of emotion expression, [17] present an approach to affective sensing, in spoken language and facial expressions, using a generic model of affective communication and a set of ontologies to assist in the analysis of concepts and the recognition process.

7 Conclusion and Future Works

In this paper, we address an original topic: how to detect conflicts between people in an e-collaborative work. The idea is to stop verbal rise in emails by the manager of an e-learning platform. We proposed to treat this problem as a classification issue. Exchanged emails are analysed to detect the birth of a conflict. We manually developed from scratch a negative emotion ontology, which has been enhanced by using the notion of triggers collected from corpora. Then with this new ontology we collected corpus related to the appropriate emotions. The corpus has been extracted from discussion forums. We got 2.138 polluted messages, which have been split into eight different emotion categories. Classification has been handled by two methods TFIDF and SVM. For this particular corpus,

we obtained good results and TFIDF outperforms SVM. Several tracks are under investigation in order to increase more easily the size of the corpus. We are also working on other classification methods. The objective is to have a list of methods in order to take advantage of each of them. In fact, our experience in topic identification showed that each classification method could succeed in the identification of some concepts and fail for others, that is why in general several classifications methods are combined to get better results.

Moreover, the possibilistic DL approach, together with the set of measures we have proposed, enable to derive new solutions based on the use cases proposed in Section 5.4.. We are currently working on an efficient interface to configure declaratively such solutions. We want to go further in the use of the model for conflict management (Figure 2) and perform content analysis of the emails, based on linguistic relations between *Fragment, Subject* and *Topic* classes. This enables us to identify the largest set of emails related to a potential conflict.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Bigi, B., Brun, A., Paul Haton, J., Smaili, K., Zitouni, I.: A comparative study of topic identification on newspaper and e-mail. In: Proceedings of the String Processing and Information Retrieval Conference, SPIRE 2001 (2001)
3. Broches, R.S.: Unraveling the Hawthorne Effect: An Experimental Artifact 'Too Good to Die'. PhD thesis, University of Wesleyan (April 2008)
4. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning, 273–297 (1995)
5. Crammer, K., Singer, Y., Cristianini, N., Shawe-taylor, J., Williamson, B.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research 2, 2001 (2001)
6. Denoual, E.: Méthodes en caractères pour le traitement automatique des langues. PhD thesis, University of Joseph Fourier (2006)
7. Dias, G.: Extraction automatique d'associations lexicales à partir de corpora. PhD thesis, University of Orleans (December 2002)
8. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D., Hogger, C., Robinson, J., Nute, D. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3, pp. 439–513. Oxford University Press, Oxford (1994)
9. Eden, L.: Multinationales en Amrique du Nord. illustrated (1994)
10. Enguehard, C., Pantera, L.: Automatic natural acquisition of a terminology. Journal of Quantitative Linguistics 2(1), 27–32 (1995)
11. Francisco, V., Gervás, P., Peinado, F.: Ontological reasoning to configure emotional voice synthesis. In: Marchiori, M., Pan, J.Z., Marie, C.d.S. (eds.) RR 2007. LNCS, vol. 4524, pp. 88–102. Springer, Heidelberg (2007)
12. Garcia-Rojas, A., Vexo, F., Thalmann, D., Raouzaiou, A., Karpouzis, K., Kollias, S.: Emotional Body Expression Parameters In Virtual Human Ontology. In: Proceedings of 1st Int. Workshop on Shapes and Semantics, pp. 63–70 (2006)
13. Haton, J., Cerisara, C., Fohr, D., Laprie, Y., Smaili, K.: Reconnaissance automatique de la parole. Du signal à son interprétation. Dunod (2006)
14. Larivery, M.: Les genres d'émotions. La lettre du psy 2(7) (July 1998)

15. Lavecchia, C., Smaili, K., Langlois, D., Haton, J.-P.: Using inter-lingual triggers for machine translation. In: Eighth conference INTERSPEECH (2007)
16. Mathieu, Y.Y.: Annotation of emotions and feelings in texts. In: Tao, J., Tan, T., Picard, R.W. (eds.) ACII 2005. LNCS, vol. 3784, pp. 350–357. Springer, Heidelberg (2005)
17. McIntyre, G., Göcke, R.: Towards Affective Sensing. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4552, pp. 411–420. Springer, Heidelberg (2007)
18. Robertson, S.E., Jones, S.K.: Relevance weighting of search terms. Journal of the American Society for Information Science 27(3), 129–146 (1976)
19. Salton, G., Mcgill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
20. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
21. Voltz, R., Oberle, D., Staab, S., Motik, B.: Kaon server - a semantic web management system. In: Alternate Track Proceedings of the Twelfth International World Wide Web Conference, Budapest, Hungary, May 2003, pp. 139–148. ACM, New York (2003)
22. Weston, J., Watkins, C.: Multi-class support vector machines (1998)
23. Whittaker, S., Sidner, C.: Email overload: exploring personal information management of email. In: CHI 1996: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 276–283. ACM Press, New York (1996)
24. Xu, L., Lin, H.: Ontology-driven affective chinese text analysis and evaluation method. In: Paiva, A.C.R., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 723–724. Springer, Heidelberg (2007)
25. Yan, J., Bracewell, D.B., Ren, F., Kuroiwa, S.: The creation of a chinese emotion ontology based on hownet. Engineering Letters 16(1), 166–171 (2008)
26. Yang, Y., Liu, X.: A re-examination of text categorization methods (1999)
27. Zacklad, M.: Communities of action: a cognitive and social approach to the design of cscw systems. In: GROUP 2003: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, pp. 190–197. ACM Press, New York (2003)
28. Zhu, B., Chen, H.: Communication-garden system: Visualizing a computer-mediated communication process. Decis. Support Syst. 45(4), 778–794 (2008)