



**HAL**  
open science

## Meeting Intents Detection Based on Ontology for Automatic Email Answering

Manon Cassier, Zied Sellami, Jean-Pierre Lorré

► **To cite this version:**

Manon Cassier, Zied Sellami, Jean-Pierre Lorré. Meeting Intents Detection Based on Ontology for Automatic Email Answering. 30es Journées Francophones d'Ingénierie des Connaissances, IC 2019, AFIA, Jul 2019, Toulouse, France. pp.99-111. hal-02329566

**HAL Id: hal-02329566**

**<https://hal.science/hal-02329566>**

Submitted on 23 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Meeting Intents Detection Based on Ontology for Automatic Email Answering

Manon Cassier<sup>1</sup>, Zied Sellami<sup>2</sup> and Jean-Pierre Lorré<sup>2</sup>

<sup>1</sup> Laboratoire AGORA, Université de Cergy-Pontoise  
manon.cassier@u-cergy.fr

<sup>2</sup> Linagora GSO, 75 route de Revel, 31500 Toulouse, France  
{zsellami, jplorre}@linagora.com

**Abstract** : Automatic email answering is a difficult AI problem that combines classification, natural language understanding and text generation techniques. We present an original approach and a tool based on an ontology to automatically reply to meeting emails. We constructed the ontology from a French corpus of 1150 emails in which the concepts represent detailed meeting intents (proposing a meeting, cancelling a meeting, rescheduling a meeting) and different answer templates. Each intent concept is a semantic rule formalized according to the FrameNet methodology. These rules are used to detect intents in emails and also to extract relevant information (such as date, time or person) used for generating replies. The main advantage of our approach is the generation of more precise answers than those proposed by other approaches. We tested the intent detection step on a set of 297 emails and compared it with different supervised machine learning algorithms. Obtained results are encouraging, with an accuracy 20% higher than results obtained with other algorithms.

**Mots-clés** : Ontology engineering, knowledge acquisition from text, knowledge-based recommendation systems.

## 1 Introduction

Automatic email answering is an interesting feature in a business context. According to the Radicati Group 2017 report<sup>1</sup> an employee receives in a day 88 emails and sends 34. Email management would represent between 5 and 10 hours of an employee's time over a month. A too massive use of emails can heavily impact work productivity by causing information overload. Automatic reply to emails therefore represents a considerable challenge.

Our work fits in a larger project to design an Open-Source collaborative platform for businesses called OpenPaaS<sup>2</sup> including a mailbox, an enterprise social network and a shared agenda. Considering the problem of email management, our goal is to propose an email processing assistant that would be able to assist the user by automatically prioritizing emails, sending notifications when urgent emails are received and generating answers.

In this paper, we present our approach for automatic email answering based on an ontology to automatically detect meeting intents in emails and to generate appropriate answers using text pattern matching and named entities detection.

The paper is organized as follows. In Section 2, we review related work. In Section 3, we detail our approach by describing the ontology and how it is used for detecting meeting intents and for generating answers. In Section 4, we compare our system to other intents detection techniques and we test the quality of the generated answers. In Section 5 we discuss the limits and possible improvements of our approach. Section 6 concludes.

---

<sup>1</sup>Report available at <https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf>

<sup>2</sup>OpenPaaS: <https://open-paas.org/>

## 2 Related Work

The problem of automatic emails answering has been studied frequently in the last ten years. (Katakis *et al.*, 2006) offer a complete view of the different approaches frequently used for email classification, automatic email summary generation or email answering. Automatically answering to an email requires two steps: detection of intents to find those that require an answer and the generation of the answer according to the detected intent.

These steps were mainly studied through different approaches, exploiting either machine learning algorithms or text-pattern matching.

(Malik *et al.*, 2007) use key phrase extraction and text similarity to perform email classification. A Naïve Bayes model automatically detects key phrases of length up to 3 words in the incoming email and maps them with question-answers pairs already identified to choose which one is similar.

(Kannan *et al.*, 2016) propose Smart Reply for Inbox by Gmail (Google) by using recurrent neural networks (RNNs) as long short-term memory (LSTM) networks to predict the most likely responses for an incoming message. Most of the answers proposed by this system are appropriate for informal context – for instance for yes/no questions. However, these answers do not correspond to the precise response model we expect for professional use. Moreover, these types of approaches require access to a considerable body of data that we do not have.

We believe that symbolic approaches can describe more precisely the elements that the answers must contain. (Carvalho, 2008) introduces the notion of Email acts inspired by the Speech Act Theory of (Austin, 1962) and (Searle, 1969). These acts are described as “*noun-verb pairs that express typical intentions in email communication – for instance, to request for information, to commit to perform a task or to propose a meeting*”. (Carvalho, 2008) proposes a taxonomy of these “Email Speech Acts” (i.e. “intents” or “intentions”) that are associated with some verbs or nouns. This taxonomy contains only general concepts<sup>3</sup>, which are not enough to describe the phenomena that we wish to process in emails. In our approach, we focus on the *meet* act which may itself be divided into several sub-intents.

(Sneiders, 2010) and (Kosseim *et al.*, 2001) propose two question-answering approaches based on symbolic rules to answer emails. These approaches use pattern matching to find specific questions in emails. For each question, a set of standard answers are defined and used on generating a draft reply. In our work, we want to detect answers, affirmations, notifications, assertions, and so on. We want to model a more complex structure than regular expressions. We use FrameNet<sup>4</sup> to formalize semantic frames to detect intents on emails.

The FrameNet project (Ruppenhofer *et al.*, 2016) gives a formalization of semantic frames as described in Charles J. Fillmore’s works (Fillmore, 1976). Each semantic frame is represented by one or more frame elements (FEs) that are evocated by words called lexical units (LUs). These frames are only checked if the frame elements are actually present in a sentence.

For example, in case of proposing an appointment by email, the semantic frame *appointment proposal* in the sentence “Je vous propose de faire une réunion jeudi prochain” (I suggest we meet next Thursday) can be detected thanks to the lexical units “propose” and “réunion” and the frame elements SPEAKER (i.e. “Je”), PROPOSITION (i.e. “propose”) and TIME (i.e. “jeudi prochain”).

Our contribution consists on building a domain ontology from a French corpus of corporate emails based on FrameNet formalization. This ontology is used to extract meeting intents from emails and for automatic emails answering.

<sup>3</sup>The taxonomy contains the 7 concepts *request*, *propose*, *deliver*, *commit*, *directive*, *commissive* and *meet*.

<sup>4</sup>The FrameNet Homepage: <https://framenet.icsi.berkeley.edu/fndrupal/> (last accessed 2018/06/08).

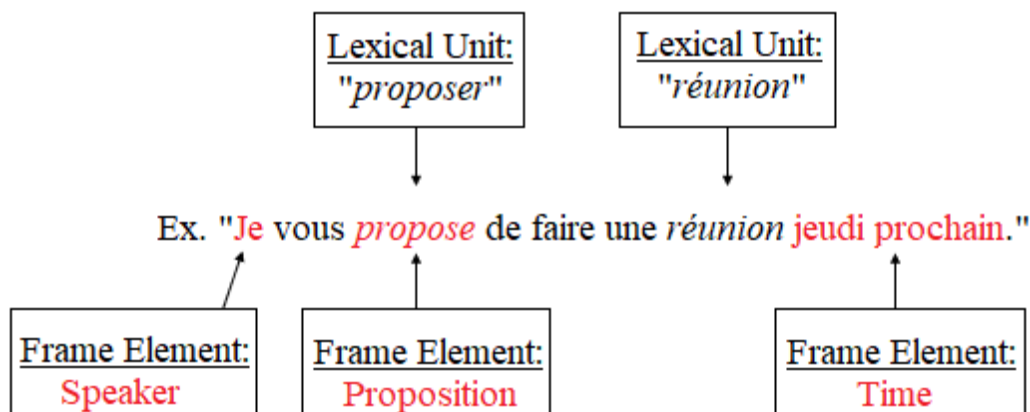


Figure 1: A Frame representation of an appointment proposal.

### 3 Ontology-based Email Answering

We manually built an OWL ontology<sup>5</sup> from a large corpus of 30000 business emails<sup>6</sup> – empty, English or social network notification emails were removed. The OWL file of the ontology is available on the project github repository<sup>7</sup>.

We used the TXM concordancer<sup>8</sup> to explore emails and quickly retrieve those containing meeting intents. This step also allowed us to extract two sets of respectively 1150 annotated emails for the ontology building (with 458 emails containing at least a meeting intent and 692 without any meeting intent) and 177 annotated emails (with 143 emails containing meeting intents) for a subsequent evaluation step.

The first level of the ontology contains 3 core concepts: the Core Intent that models meeting intents, the Core Answer concept that models answer patterns for intents and the Frame Element Concept that models frame elements classes (e.g Time, Date, Address, Person).

#### 3.1 Meeting Intent Concepts

We identified 18 specific meeting intent concepts organized on 3 main concepts (Request, Proposal and Notification) as presented in table 1.

Each intent concept contains lexical units and frame elements (which are Annotation Properties in the ontology) that allow detecting a specific intent in an email. A lexical unit is defined with a String value (lemma form), a part-of-speech tag (e.g. VERB, NOUN, ADJ) or a regular expression (EXPR) and an Annotation Value (Mandatory or Optional). A frame element is defined with a Frame Element Class resource (these are generally what we detect as named entities) or a Frame Element Individual resource – which are defined by a regular expression containing words, expressions or characters that evocate them – and an Annotation Value (Mandatory or Optional). For instance, the *request* concept contains the frame element INTERROGATION – represented by a regular expression containing the strings “?”, “est-ce que”, “quel”, etc. (i.e. the English WH questions words equivalents) – and the lexical units “demander” (ask for), “est-ce possible” (is it possible), “(jelnous) souhait(elons)” (I/we wish), etc. The Annotation Value determines if a lexical unit or a frame element is necessary

<sup>5</sup>We build the ontology with the Protégé tool (Stanford Center for Biomedical Informatics Research, 2016): <http://protege.stanford.edu/>.

<sup>6</sup>The corpus can not be distributed for confidentiality reasons.

<sup>7</sup><https://github.com/openpaas-ng/automatic-email-answering/blob/master/intent6.owl>

<sup>8</sup>TXM Concordancer (Heiden, 2010): <http://textometrie.ens-lyon.fr/>.

or not for identifying the intent on the email (i.e. if the element is specific to the current intent or not). This information helps to compute a score when using the ontology to automatically annotate emails.

Table 1: Description of the core intents of the meet act ontology.

<i>Request</i>	The <i>request</i> concept includes both requests for information and request addressed to the recipient to perform some activity related to an appointment. It contains 7 intents: <i>request an appointment cancellation</i> , <i>request an appointment confirmation</i> , <i>request to schedule an appointment</i> , <i>request an appointment change</i> , <i>request details about an appointment</i> , <i>request availabilities for an appointment</i> and <i>request a participation to an appointment</i> .
<i>Proposal</i>	The <i>propose</i> concept includes proposals addressed to the recipient to do something or to take part in something related to an appointment. It contains 3 intents: <i>propose an appointment</i> , <i>propose an appointment cancellation</i> and <i>propose an appointment change</i> .
<i>Notification</i>	The <i>notification</i> concept includes all the messages that just observe a fact about an appointment. It contains 8 intents: <i>availability confirmation</i> , <i>appointment confirmation</i> , <i>appointment cancellation notification</i> , <i>unavailability notification</i> , <i>availability notification</i> , <i>appointment reminder notification</i> , <i>precisions about an appointment</i> and <i>appointment change notification</i> .

Shared frame elements and lexical units are described in the 3 main concepts so that sub-concepts that represent intents inherit them. For example, the frame element Interrogation is transmitted to all intents related to the main concept request differentiating them from intents related to other concepts propose and notification.

A higher concept called meeting containing only lexical units that refer to an appointment (e.g. “entretien”, “RDV”, “reunion”) is used to filter the incoming emails and detect only those containing sentences related to an appointment.

The ontology is enriched by non-hierarchical relations between intents (called implication relation) that model associations of intentions within a same email. For example, in the sentence “Je ne suis pas disponible demain, pouvons-nous décaler l’entretien ?” (I’m not available tomorrow, can we change the time of my interview), unavailability notification intent implies an appointment change request.

### 3.2 Answer Pattern Concepts

For each meeting intent, we modeled in the ontology a set of answer templates (patterns). These templates were determined by analyzing the responses that people give to each type of appointment email. An answer can be shared between multiple intents.

For example, a template for *accepting meeting proposal* is: « D’accord pour {[DET;values=le,la,l’;default=le] [Appointment\_Intent;default=rendez-vous]} {[Time]} {[DET;values=avec;default=avec] [Person;default=]} ». In this template, Time, Person and Appointment\_Intent will be changed with corresponding values recovered from the sentence in the email that instantiated the *accepting meeting proposal* intent.

Note that our system will be able to use the calendar module to check when the person is available for an appointment. This information will be also used to generate a more precise answer.

### 3.3 The System

Our system is integrated to the OpenPaaS UnifiedInbox module (Figure 2) that communicates with our answers suggestion service through a REST API. The system is open-source and

available on github<sup>9</sup>.

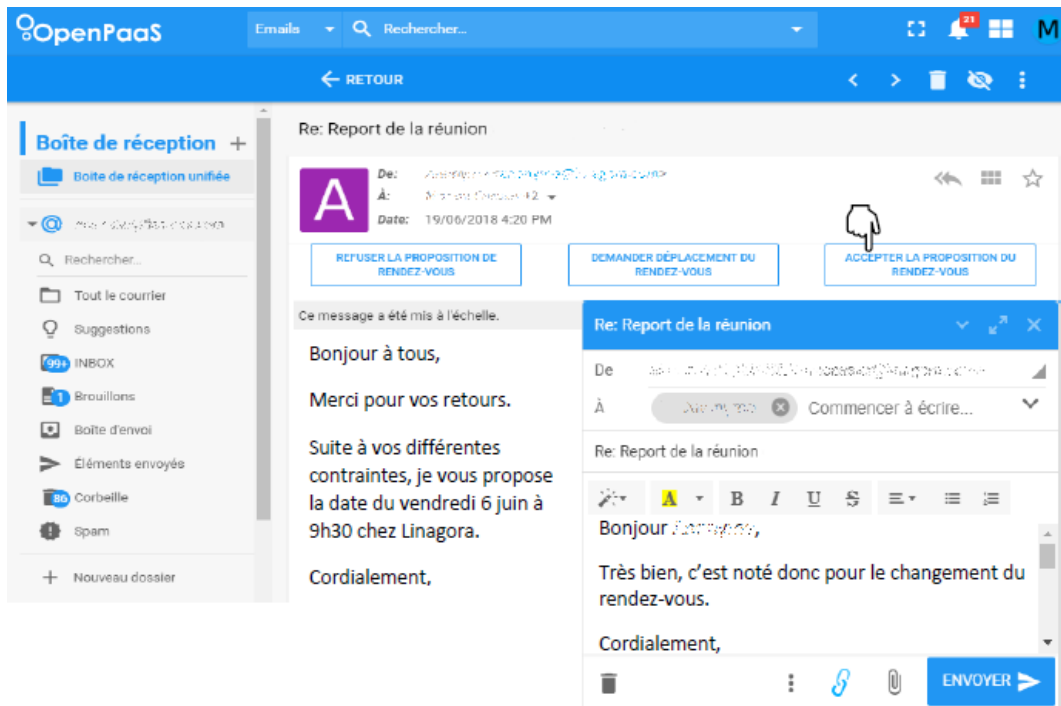


Figure 2: Screenshot of an example of answers suggestions in OpenPaaS.

For each new email, a JSON request is sent to the Web Service module with the text of the email and other metadata (sender, recipients, date). The system analyzes the query and returns a JSON response with answer suggestions as presented in Figure 3.

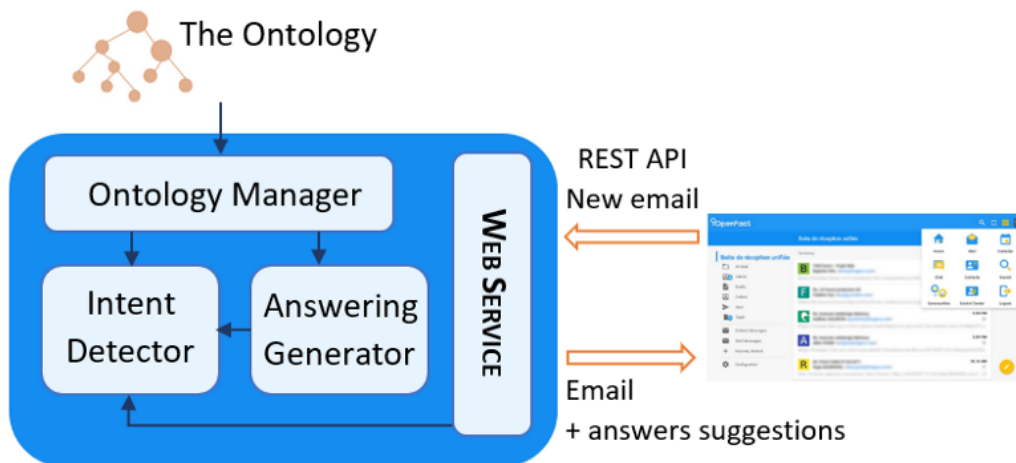


Figure 3: General architecture of the automatic email answering system.

Our system has four main modules:

1. An **Ontology Manager** that uses the Apache Jena API to parse and load the ontology in a Java object.

<sup>9</sup><https://github.com/openpaas-ng/automatic-email-answering>

2. An **Intent Detector** that integrates the CoreNLP<sup>10</sup> (Manning *et al.*, 2014) tool which makes tokenization, POS tagging and named entities detection on the text of the email. It exploits at the same time the Duckling Facebook<sup>11</sup> tool that is more successful for the detection of encrypted named entities. When two different named entities are proposed, a resolver automatically selects the proposal made by Duckling Facebook. The text of the email and the lexical units contained in the ontology are then completely stemmed<sup>12</sup> by the Snowball Stemmer<sup>13</sup> to allow easier matching of forms with text shapes. The module retrieves the information available in the ontology and makes a projection of the lexical units of the higher intent meeting on the text of the email to check if the current email contains meeting intents or not. This projection is made on the entire message thread to which the target email belongs to avoid missing emails that do not contain the word “rendez-vous” (appointment) or a synonym but do contain an intention. For example, it allows conserving the sentence “Oui, je suis disponible” (Yes, I’m available) answering to “Êtes-vous disponible demain pour un rendez-vous ?” (Are you available tomorrow for a meeting?). If no intent is detected, the system does not propose any intent and the Web Service returns the message meeting intent not detected. Otherwise, it makes a projection of the other lexical units and frame elements only on the sentences of the incoming email (without taking the message thread or the order of appearance of LUs and FEs in account) and computes a score by assigning a higher weight to units marked as mandatory in the ontology to select the three more likely intents – after several experiments, we also chose to give a higher score to lexical units than frame elements. Finally, the module computes an annotation score (described below) to each intent detected thanks to the matching step.
3. An **Answering Generator** that generates an answer for each intent detected. This module integrates an email generator that recovers the answers templates associated to detected intents. It adds a greeting according to the time – for example, “Bonsoir” (Good evening) after 8:00 pm – and a closing formula as “Cordialement” (Sincerely). It computes a score for each answer, corresponding to the sum of the instances scores of each intention in the email. This module will also exploit the user’s calendar to check its availabilities and complete the Time, Person and Appointment\_Intent tags in the answers templates.
4. A **Web Service** that returns the three first associated answers in a JSON format.

The annotation score is computed after a filtering of the matching results. The system only keeps concepts of the ontology lowest level (i.e. the intents) that match at least two instances of lexical units or frame elements and of which at least one of the lexical units matched is mandatory.

The score assigned to an intention corresponds to its relevance value (1) in relation to the sentence studied. It corresponds to the sum of its annotation value (2) and its specificity value (3) as described below.

$$\text{Relevance value} = \text{Annotation value} + \text{Specificity value} \quad (1)$$

$$\begin{aligned} \text{Annotation value} = & nbInstances + nbSpecificFE \\ & + 4 * nbSpecificLU \\ & + 4 * nbSpecificMandatoryInstances \\ & + nbMandatoryInstancesOfSuperConcept \end{aligned} \quad (2)$$

<sup>10</sup>CoreNLP: <https://stanfordnlp.github.io/CoreNLP/>

<sup>11</sup>Duckling Facebook: <https://duckling.wit.ai/>

<sup>12</sup>The choice of a stemmer is motivated by the fact that we did not find an open source lemmatizer effective enough for French.

<sup>13</sup>SnowballStemmer: <https://snowballstem.org/>

$$\text{Specificity value} = \text{Ratio 1} + \text{Ratio 2} + \text{Ratio 3} + \text{Ratio 4} \quad (3)$$

$$\text{Ratio 1} = \left( \begin{array}{l} \sum [\frac{1}{nbMandatoryLU}] \text{ if the mandatory LU is instanciated} \\ \sum [\frac{-1}{nbMandatoryLU}] \text{ if the mandatory LU is not instanciated} \end{array} \right) \quad (4)$$

$$\text{Ratio 2} = \left( \begin{array}{l} \sum [\frac{1}{nbMandatoryFE}] \text{ if the mandatory FE is instanciated} \\ \sum [\frac{-1}{nbMandatoryFE}] \text{ if the mandatory FE is not instanciated} \end{array} \right) \quad (5)$$

$$\text{Ratio 3} = \sum [\frac{1}{nbOptionalLU}] \text{ if an optional LU is instanciated} \quad (6)$$

$$\text{Ratio 4} = \sum [\frac{1}{nbOptionalFE}] \text{ if an optional FE is instanciated} \quad (7)$$

The parameters of the annotation value are obtained empirically. An instance of an intent is correct if it is instantiated with mandatory and specific lexical units or frame elements.

The specificity value is obtained by calculating the ratio of frame elements and lexical units. This formula makes it possible to estimate the level of coverage of the instance in relation to the properties of its class.

We detail next how these scores are calculated for an intent of *request availabilities for an appointment* in the sentence "Pouvez-vous m'indiquer vos disponibilités pour planifier un rendez-vous?" (Please let me know when you're available). The different parameters used to compute the annotation value of the sentence are:

- Number of Instances = 4 ("pouvez-vous", "disponibilités", "planifier" and "rendez-vous")
- Specific FE = 0
- Specific LU = 2 ("disponibilités" and "planifier")
- Specific mandatory instances = 1 ("disponibilités")
- Inherited mandatory instances = 2 ("pouvez-vous" (*request intent*) and "rendez-vous" (*meeting intent*))

$$\text{Annotation value} = 4 + 4 * (2 + 1) + 2 = 18$$

The *request availabilities for an appointment* contains 14 mandatory LU, 4 optional LU, 1 mandatory FE and 1 optional FE. We calculate the specificity value with the results of the 4 ratios.

$$\text{Ratio 1} = \frac{1}{14} + 13 * \frac{-1}{14} \quad \text{Ratio 2} = \frac{-1}{1} = -1 \quad \text{Ratio 3} = \frac{1}{4} = 0.25 \quad \text{Ratio 4} = 0$$

The specificity value of the intent *request availabilities for an appointment* for the sentence is:

$$\text{Specificity value} = -0.86 - 1 + 0.25 = -1.61$$

Finally we get the relevance value of the intent to the sentence:

$$\text{Relevance value} = 18 - 1.61 = 16.39$$



## 4 Performance Measurements

We tested the performance of the system with 177 annotated emails. Each email in the corpus was classified into different folders representing intents. For example, an email containing an intention to confirm an appointment and request details about an appointment was filed in two folders named with the intent title. Thus, the 143 emails that contain intents in our corpus offer a total of 297 occurrences of the different intents.

We evaluated our system according to two criteria: its performance in email classification and the relevance of the answers it proposes. We made the same evaluation with other systems based on machine learning algorithms.

### 4.1 Effectiveness in Intents Detection

To compare our approach to machine learning approaches, we trained machine learning models with the 1150 annotated emails used to build the ontology and we tested them on the 177 emails corpus. We applied a Cross Validation method on predictive models using Logistic Regression, Decision Tree, Random Forest and Naive Bayes algorithms with a Bag of Words approach scored with TF-IDF. The Cross Validation suggests a best model based on Random Forest algorithm for the intent detection task.

We calculated precision, recall and f-score of the two systems based on the first intent proposition and the three intent propositions.

Table 2 and 3 show respectively the results of our system and of the machine learning based model.

Table 2: Precision, recall and f-score obtained by the ontology-based system.

	Precision	Recall	F-score
According to the 1st intent	0.50	0.48	0.49
According to the three intents	0.71	0.67	0.69

Table 3: Precision, recall and f-score obtained by the machine learning based model.

	Precision	Recall	F-score
According to the 1st intent	0.25	0.17	0.20
According to the three intents	0.25	0.23	0.24

For each evaluation step, our system presents far better results than the machine learning based model. In addition, we notice that our system captures more easily distinctions between the different intents (despite a decrease in the f-score caused by some confusion between intentions that share the same FEs and LUs) while machine learning has difficulties making multiclass distribution.

Table 4 shows the percentage of instances correctly identified by both systems for each intent. We compute these ratios according to the three proposed intents by each system since it is more likely to find the right intention on three proposals.

Our ontology-based system obtains the best results for almost all categories. The low results obtained by the machine learning system for all the categories can be explained by the size of the training corpus. Such a system requires many annotated examples to be able to learn and classify intents on its own. This experience shows instead that our approach is viable even with a small set of data.

The machine learning system obtained the best score on detecting the *precisions about an appointment* intent. It can be explained by the fact that many emails of the corpus contain this intent. A closer look at the results shows that the system tends to classify almost all emails with this intent. It is therefore logical that the proposed answer is often correct for this case.

Table 4: Effectiveness of intent detection for each intent type.

Intents	Number of instances to find	Ontology based results	Ontology based ratio	ML based results	ML based ratio
Request an appointment cancellation	2	2	<b>100%</b>	0	0%
Request an appointment confirmation	22	16	<b>73%</b>	13	59%
Request to schedule an appointment	8	3	<b>38%</b>	0	0%
Request an appointment change	3	0	0%	0	0%
Request details about an appointment	10	9	<b>90%</b>	1	10%
Request availabilities for an appointment	25	20	<b>80%</b>	6	24%
Request a participation to an appointment	7	3	<b>43%</b>	0	0%
Propose an appointment	39	30	<b>77%</b>	29	74%
Propose an appointment cancellation	2	1	<b>50%</b>	0	0%
Propose an appointment change	11	7	<b>64%</b>	0	0%
Availability confirmation	4	2	<b>50%</b>	2	<b>50%</b>
Appointment confirmation	8	4	<b>50%</b>	2	25%
Appointment cancellation notification	7	4	<b>57%</b>	0	0%
Unavailability notification	36	23	<b>64%</b>	6	17%
Availability notification	3	2	<b>67%</b>	1	33%
Appointment reminder notification	25	16	<b>64%</b>	3	12%
Precisions about an appointment	47	35	75%	42	<b>89%</b>
Appointment change notification	4	1	<b>25%</b>	0	0%

Anyway, our system seems able to correctly classify most of the intents. We notice that it does not identify the *request an appointment change* intent that is probably too close to the *propose an appointment*. In many cases, our system suggests a *propose an appointment* intent instead of a *request an appointment change* intent. Moreover, this case is not really a problem as long as given that the two intentions share similar answer templates. We do not combine these two intents into one because the *request* concept and the *proposal* concept are semantically different.

## 4.2 Evaluation of the Relevance of the Proposed Answers

We compared the relevance of the answers proposed by our system and by Google’s Smart-Reply (Kannan *et al.*, 2016) – that uses Deep Learning and LSTM neural networks to propose 3 answers – for 15 emails. The relevance of the answers was evaluated by 26 Linagora employees through a questionnaire.

For each email, we asked annotators to choose between two blocks of answers – the Google’s one and ours – the one that they felt contained the most relevant answer. When none of the proposed answers were relevant to the email submitted, annotators could select an option “no relevant answer”. However, we did not allow annotators to select proposals from both systems at the same time even though the proposed answers were all relevant. We wanted to get as close as possible to a real situation where the user of an automatic answering system must necessarily choose one answer between several choices. We did not indicate which answers were proposed by the Google’s system or by ours to obtain the most objective results possible.

Table 5 details the obtained results from 26 annotators by explaining the intents contained in each email submitted for annotation.

*Table 5: Distribution of the chosen answers for each email.*

Email	Contained intent	Ontology Based System	Google’s Smart Reply	No relevant answer
1	Propose an appointment Request availabilities for an appointment	<b>15</b>	5	6
2	Request a participation to an appointment	5	<b>19</b>	2
3	Appointment confirmation	7	<b>18</b>	1
4	Request an appointment confirmation	<b>24</b>	1	1
5	Propose an appointment Request availabilities for an appointment	<b>16</b>	8	2
6	Propose an appointment cancellation	9	<b>17</b>	0
7	Precisions about an appointment	4	<b>20</b>	2
8	Appointment confirmation Request an appointment confirmation	5	<b>20</b>	1
9	Propose an appointment change	<b>13</b>	10	3
10	Propose an appointment cancellation	<b>20</b>	0	6
11	Request details about an appointment	<b>21</b>	5	0
12	Unavailability notification	<b>12</b>	<b>12</b>	2
13	Propose an appointment Availability notification Request availabilities for an appointment	6	<b>17</b>	3
14	Propose an appointment Request an appointment confirmation	<b>22</b>	4	0
15	Appointment cancellation notification	<b>23</b>	2	1
<b>Total:</b>		202	158	30

The table shows that the answers obtained by our system are preferred for 8 emails out of 15 with an agreement of more than 75% for 5 emails against 7 emails out of 15 for the Google’s system – with an agreement of more than 75% for only 2 emails. On the 12th mail, the same number of annotators chose the answers of the two systems. The total number of annotators’ choices for responses from our system (202) is higher than Google’s responses (158). We can also notice that our answers are still chosen by more than two annotators in all the cases for which the answers proposed by Smart-Reply are mainly selected by the annotators. In all cases, the answers proposed by our system were chosen by at least one

annotator which proves that these answers are grammatically correct. This experience shows that our approach allows us to provide answers that are relevant and acceptable to users. The results show that our answer proposals can fully compete with those proposed by the Google's system. A closer look at the annotators' choices also shows that the user prefers the most complete answers possible – that are proposed by our system. These are particularly chosen when the answer requires a mention of specific elements such as availabilities or precisions about an appointment. Figure 4 illustrates an example of sentence mainly chosen by annotators for an email containing an *appointment confirmation request* intent.

**Received email:**  
"Re bonjour, je vous ai envoyé une invitation d'entretien pour mercredi 23 de 8h30 à 9h. Merci de bien vouloir la confirmer. "  
(Good morning again, I sent you a meeting invitation for Wednesday the 23<sup>rd</sup> from 8:30 a.m. to 9:00 a.m. Thanks for confirming.)

**Ontology-based system answers (24 choices):**

1. "D'accord pour faire le rendez-vous mercredi 23 de 8h30 à 9h." (Okay to make the appointment Wednesday 23<sup>rd</sup> from 8:30 a.m. to 9:00 a.m.)
2. "Je ne serai malheureusement pas disponible pour ce créneau. Pouvons-nous trouver un autre créneau pour le rendez-vous ?" (Unfortunately, I will not be available for this time slot. Can we find another time slot for the meeting?)
3. "Je ne pourrai pas me libérer pour le rendez-vous mercredi 23 de 8h30 à 9h." (I won't be able to make an appointment Wednesday 23<sup>rd</sup> from 8:30 a.m. to 9:00 a.m.)

**Google's Smart-Reply answers (1 choice):**

1. "Bien reçu." (Alright)
2. "D'accord." (Okay)
3. "Bien reçu, merci." (Alright, thanks)

**No relevant answer (1 choice)**

Figure 4: Sample annotation for an email.

## 5 Advantages, Limitations and Further Research

Automatic email answering is a large studied problem. We propose a new way to resolve this problem by using an ontology to detect intent on emails and answer to them. Unlike machine learning solutions like Google's Smart-Reply our approach does not need large data set to run. Using an ontology is also an effective way to model complex semantic facts from text like meeting intents. Our system proposes answers more precise – thanks to the detection of complex intents – and complete with details on time, date or place of the meeting – thanks to named entities detection. Our approach is generic and new rules can easily be added to the ontology to detect new intents and answer to them. In our opinion, such a system is more controllable and understandable than a machine learning solution.

However, our approach has some limitations since the addition of intent concepts can be costly in time and thought. If the rules described by an intent concept are too precise, the system can easily merge several intents and provide noise. On the contrary, the system risks

missing many occurrences and thus generating silence if the rules associate too few lexical units and frame elements.

In addition, the effectiveness of this kind of symbolic rules depends on several factors:

- the domain: for instance, the meeting intents can be detected with the lexical units “table ronde” (a round-table discussion) or “colloque” (a symposium) in an academic setting, but rather with the lexical units “réunion” (a meeting) or “point” (a point) in a professional setting.
- the spelling: as the OpenPaaS mailbox does not integrate a spell checker, some lexical units may not be detected if misspelled. Moreover, it is difficult to manage all variations of declensions without an effective lemmatizer for French, which has many irregular verbs.
- synonym management: the rules must account for as many synonyms as possible to be able to recognize them all. A non-identified synonym cannot be recognized by the system.

We plan to improve our work by enriching the ontology with other intents (recruitment intents, reminder intents, document sharing intents, etc.). We’ll improve the generation of email answers by detecting situations that require a formal form of address in emails – for example, when the user must address a hierarchical superior. We also plan to link the answering mechanism to the user’s electronic agenda to automatically check availabilities dates and generate answers depending on the user’s constraints.

Also, it seems important to us to find a way to automatically manage lexical units synonymy – by using synonym dictionary like WordNet for instance – and to integrate a French lemmatizer to improve the detection of lexical units and frame elements in emails.

Finally, it is possible to adapt the tool to detect intents in the other languages available in CoreNLP (English, German, French, Chinese and Spanish). At the ontology level, the user will have to add the POS properties of the new language and enrich the ontology with LUs and EFs in this language. Currently, the Java code support French and English. To process a new language, we need to integrate in the Intent Detector module the CoreNLP pipeline associated with this new language. The Answering Generator module requires a lemmatizer adapted to the language. It will be necessary to modify it according to the chosen language.

## 6 Conclusions

In this paper, we presented an ontology-based approach for automatic email answering. The ontology was manually built from a corpus of French emails. It models meeting intent concepts using the FrameNet principles and different answer templates for each intent. The automatic email answering system contains four main modules to process the incoming email, spot intents with pattern matching, compute a score according to rules described in the ontology and propose answers.

We evaluated the effectiveness of the intents detection by calculating precision, recall and f-score measurements and compared the results with those obtained by machine-learning approaches. We also evaluated the relevance of the proposed answers by them with those proposed by the Google’s Smart-Reply system.

In both cases, we get encouraging results with our system despite a limited annotated corpus. These results show that our approach is a suitable alternative to machine learning techniques (especially DNN) which are very demanding in annotated data.

We plan to enrich the ontology with other kinds of intents. We will improve the answering generation mechanisms by detecting the gender of the recipient and the T-V distinction – a linguistic formality in the French language which refers to “tu” and “vous” usages. A synonym dictionary and a lemmatizer will also be integrated to improve the matching of lexical units and frame elements in emails.

## References

- AUSTIN J. L. (1962). *How to do things with words: The William James Lectures delivered at Harvard University in 1955*. Oxford: Urmson.
- CARVALHO V. R. (2008). *Modeling Intention in Email*. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science.
- FILLMORE C. (1976). Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, volume 280, p. 20–32.
- KANNAN A., KURACH K., RAVI S., KAUFMAN T., MIKLOS B., CORRADO G., TOMKINS A., LUKÁCS L., GANEA M., YOUNG P. & RAMAVAJJALA V. (2016). Smart reply: Automated response suggestion for email. In *Conference on Knowledge Discovery and Data Mining (KDD)*. San Francisco, USA.
- KATAKIS I., TSOUMAKAS G. & VLAHAVAS I. (2006). Email mining: Emerging techniques for email management. In A. VAKALI & G. PALLIS, Eds., *Web Data Management Practices: Emerging techniques and Technologies*, p. 219–240. Idea Group Publishing, USA.
- KOSSEIM L., BEAUREGARD S. & LAPALME G. (2001). Using information extraction and natural language generation to answer e-mail. *Data Knowledge Engineering*, vol.38, p. 85–100.
- MALIK R., SUBRAMANIAM L. V. & KAUSHIK S. (2007). Automatically selecting answer templates to respond to customer emails. In *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, p. 1659–1664.
- MANNING C. D., SURDEANU M., BAUER J., FINKEL J., BETHARD S. J. & MCCLOSKEY D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.*, p. 55–60.
- RUPPENHOFER J., ELLSWORTH M., PETRUCK M., JOHNSON C., BAKER C. & SCHEFCZYK J. (2016). Framenet ii: Extended theory and practice. Available online: [https://framenet.icsi.berkeley.edu/fndrupal/the\\_book](https://framenet.icsi.berkeley.edu/fndrupal/the_book).
- SEARLE J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*, volume 626. Cambridge University Press.
- SNEIDERS E. (2010). Automated email answering by text pattern matching. In H. LOFTSSON, E. RÖGNVALDSSON & S. HELGADÓTTIR, Eds., *Proc. 7th International Conference on Natural Language Processing (IceTAL), August 16-18, Reykjavik, Iceland, LNAI 6233*, p. 381–392. Springer, Berlin, Heidelberg.