



SIXTH FRAMEWORK PROGRAMME
PRIORITY 2
Information Society Technology

LUNA

spoken Language Understanding in multilinguAI communication systems

Project n. 33549

D5.5 – Evaluation of components and their combination

Due date of deliverable:	August 3, 2008		
Actual submission date:	October 20, 2008		
Start Date of Project	September 4, 2006	Duration	36 months
Organization name of lead contractor for this deliverable	CSI		
		Version	1.0
Dissemination level	[RE]		

Authors and reviewers tables

Authors	Company Name
Stefan Hahn	RWTH
Geraldine Damnati	FT
Frederic Bechet	UAPV
Giuseppe Riccardi	UT
Marco Dinarelli	UT
David Horowitz	UT
Agnieszka Mykowiecka	PAS
Malgorzata Marciniak	PAS
Ryszard Gubrynowicz	PJIIT
Krzysztof Marasek	PJIIT

Reviewers	Company Name
Renato De Mori	UAPV
Aldo Poma	CSI

Abstract

The architecture of a software platform for a multimodal dialogue system is presented together with an application to a problem solving service.

A system for spoken opinion analysis developed for performing surveys about the satisfaction of telephone services is described and evaluation results are provided.

Systems for frame annotation of transcribed and spoken sentences are introduced.

TABLE OF CONTENTS

1	Introduction.....	5
2	Spoken Dialog System Prototypes.....	5
2.1	Italian Prototype.....	5
3	Evaluation of a LUNA SLU system with real end-users: the Opinion Mining experiment.....	9
3.1	Context of this study.....	10
3.2	Telephone survey corpora.....	11
3.3	Alarm detection method.....	11
3.4	Evaluation.....	13
4	Other results.....	15
4.1	Imposing semantic coherence.....	15
4.2	Frame annotation in Polish.....	17
5	References.....	18

1 Introduction

This deliverable describes the architecture of two systems, one for Italian and another for French which perform Spoken Language Understanding (SLU) with real-world users.

The Italian prototype developed at UT includes an SLU component into a platform for multimedia dialogues. An application is described about a problem solving activity with casual users of a service at CSI Piemonte.

The French prototype, developed in cooperation between FT and UAPV is about a telephone service for performing surveys of user satisfaction. It includes the extraction of opinion categories and polarities from spoken messages in which users describe their satisfaction about the solution of previously proposed problems. Opinions are expressed about the quality of service, the attention of the operators and other things.

Some results are also presented on the generation of frames interpreting annotations of spoken turns made by humans and automatic speech recognition systems.

Details of some system components are not reported here since they can be found in deliverables D2.2 and D3.2.

2 Spoken Dialog System Prototypes

2.1 Italian Prototype

The LUNA prototype was developed in Italian. It illustrates an opening How may I help you? prompt that allows the caller to specify the type of problem. The system requests the attribute of hardware problem and moves on to ask the make of the hardware. The prototype is comprised of an IVR platform running a speech server (VoxNauta); a communications platform that coordinates the data flow through the system and finally an expert system and knowledge base (CLIPS) that encapsulates the rules associated with dialog moves. The prototype illustrates an enhanced call routing application in the context of the HW/SW technical help desk. The enhancements comes from the use of robust SLU that segments and labels the spoken utterance with concept labels and attributes along with their confidence scores and alternate interpretations (n-best or word lattices) in real time [Moschitti A., Riccardi G. and Raymond C, 2007]. The goal of the SDS prototype is to identify problem and its attributes.

The following figure describes the architecture of the system:

Central to the system is the dialogue manager which is responsible for invoking the expert system to plan the next dialog move as will be further discussed below. In addition, the ability to store and query large amounts of data is a key requirement for data-driven dialog systems, in which the data is generated by the spoken dialog system (SDS) components (spoken language understanding (SLU), dialog management (DM), natural language generation (NLG) etc.) and the world it is interacting with (news streams, ambient sensors etc.).

The database in LUNA stores heterogeneous types of information at various levels of description in a uniform way. This uniform storage requires some upfront investment in terms of organizing the data used by the system. However, this pays off due to the ability to query the evolving data at any time in a uniform way, e.g. by performing queries across various types of information.

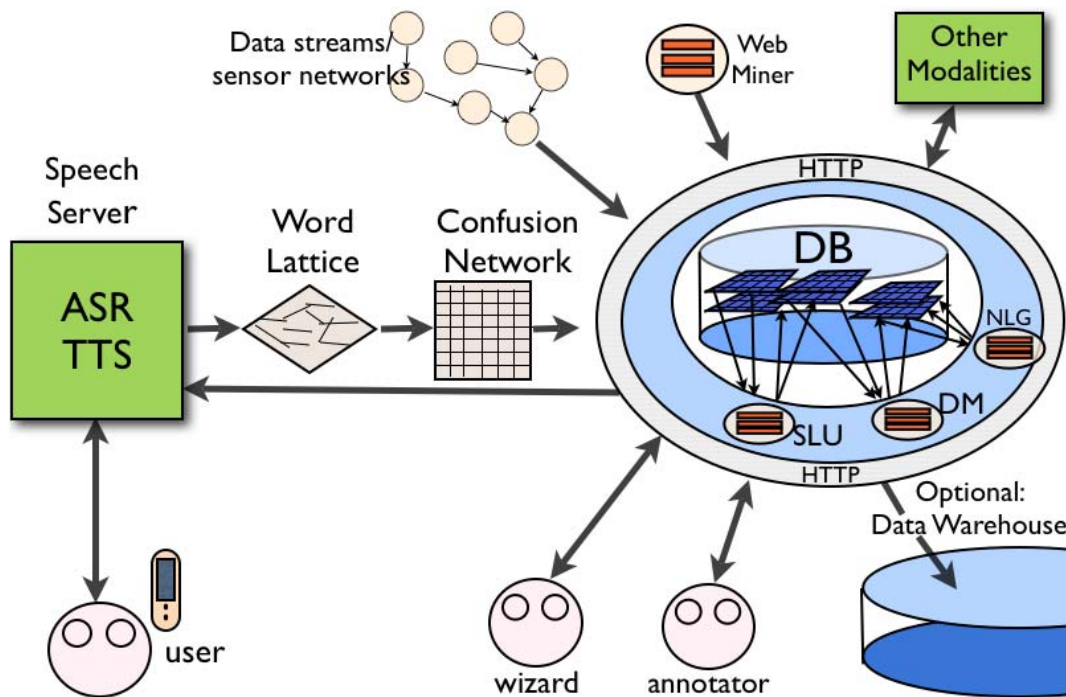


Figure 1 – Architecture of the Italian prototype

The architecture facilitates dialog evaluation, data mining and online learning because data is readily available for querying as soon as it has been stored, independent of concurrent data access by other modules. Multiple systems/applications can be made available on the same infrastructure due to a clean separation of its various aspects: processing modules (SLU, DM, NLG etc.), data storage and persistency (DBMS), and monitoring/analysis/visualization and annotation tools for human usage/consumption. Infrastructure, data storage and monitoring tools are shared among the SDS. There is no need for separate ‘logging’ mechanisms: the state of the dialog system is contained in the database, and is therefore available for analysis after the dialog ends. The dialog model is based on the Information State Update approach, using the database as its persistent ‘whiteboard’ [Varges S. Riccardi G. and Quarteroni S., 2008].

A typical interaction is initiated by a phone call that arrives at a telephony server which routes it to a VXML platform. All communication between the above-mentioned components is stored in the DBMS: ASR recognition results, TTS parameters and ASR recognition parameters reside in separate tables. The dialog manager uses the basic tables as its communication protocol with ASR and TTS engines, and additionally stores its Information State (IS) in the database. This means that the IS is automatically persistent, and that dialog management becomes a function that maps ASR results and old IS to the TTS and ASR parameters and a new IS.

The infrastructure and dialogue manager provides a platform for extending the design of the LUNA dialogs. The LUNA prototype illustrated the integration of the SLM and SLU modules and implements recognition of open spoken language utterances. The remaining part of the dialog illustrates a functioning dialog manager. A future version of a dialog will involve conversational interaction, not only in the opening prompt, but also in the collection of attributes and implementation of dialogue acts such as confirmation. Finally, the SLM will be integrated throughout the dialog so that open prompts may be invoked.

The major effort of the LUNA prototype was to develop a platform, dialog manager and expert system to promote the development of conversational dialogs. The software is tested as illustrated by the LUNA prototype. The expert system provides a handle on the interpretation of spoken language utterances. The system has been designed to scale to more complex dialogs through the

addition of new rules. Future work will focus on new conversational dialogs and the implementation of new dialog acts.

The logic that the DM embodies is coded in a knowledge base/expert system. The expert system allows for a compact representation as well as inference on facts and rules. The rule-base implemented in CLIPS, a LISP-like expert systems language forms the core logic of the dialog manager. A specific example of a rule in the DM is the confirmation dialog act and it is discussed in what follows. The point is to illustrate how the DM makes decisions about events occurring in the LUNA system, such as low confidences and how to handle confirmations and error recovery. The following code excerpt illustrates the logic of the confirmation dialog act (specifically implicit confirmation) and is written in pseudo-code for readability.

```
IF (class-confidence <= upper-threshold-confidence-problem-class) AND (class-confidence >=
implicit-threshold-confidence-      problem-class)
THEN (assert(implicit-verification-required (parameter problem-class)(value ?class-name))
```

1. The SLU module of the prototype is implemented with AT&T FSM/GRM tools and SRILM toolkit. The model used for SLU is a Conceptual Language Model (CLM) trained using SRI tools and then converted in an FST. As described in Work Package 2, the module takes as input the ASR output (n-best or lattice) and produces the best concept annotation given a word sequence. The results of SLU module are stored in the database. In particular, each concept output by the SLU module is stored together with the word constituents and the confidence score, computed as the geometric mean of word level confidence scores provided by the ASR module.
2. The DM retrieves SLU results from the database and, performing keyword spotting on concept chunking, makes guess on which the class the user problem fits in is. In order to guess the problem class, the DM exploits also the confidence measure provided by the SLU module. Depending on the value of the confidence, the DM can just continue the dialog or ask for user confirmation. There are both explicit and implicit confirmations depending on confidence level.
3. What it follows is a list of the problem classes modelled for the task of the prototype and a list of the most common concepts:

Scenarios
<i>1. Printer problem</i>
<i>2. PC Network problem</i>
<i>3. Slow computer</i>
<i>2.2. Screen/Video card problem</i>
<i>5. Keyboard problem</i>
<i>6. Mouse problem</i>
<i>7. Office network problem</i>
<i>8. Virus issues</i>
<i>9. CD-ROM reader problem</i>
<i>10. PC power problem</i>

Table 1 - List of the problem classes (dialog scenarios) modeled in the LUNA WOZ dialogs

SLU concept tags
<i>azione</i>
<i>computer-componenteHardware</i>
<i>negazione</i>
<i>codice-valoreDiCodice</i>
<i>concetto</i>
<i>numero-cardinale</i>
<i>ubicazione-istituzione</i>
<i>problema</i>
<i>persona-nome</i>
<i>computer-componenteSoftware</i>

Table 2 - A list of the most frequent concepts used by the SLU module

An example of dialog is given below:

URNS	ATTRIBUTE CHUNKING
S: Buongiorno sono Paola, coma la posso aiutare? U: Buongiorno, ho [un problema] [col mio computer]	Interpretation Confidence: 0.8 Concetto[un problema] Confidence: 0.7 Computer-componenteHardware[col mio computer] Confidence:0.9
S: Puoi dirmi qual' e' il problema che hai riscontrato? U: [Il mio monitor] [non] [funziona]	Interpretation Confidence: 0.6 Computer-componenteHardware[Il mio monitor] Confidence: 0.9 Negazione[non] Confidence: 0.2 Azione[funziona] Confidence: 0.7
S: Forse il tuo monitor e' solo spento ? U: [No], l' [ho acceso], ma [non] [funziona] comunque	Interpretation Confidence: 0.8 Negazione[No] Confidence: 0.7 Azione[ho acceso] Confidence: 0.9 Negazione[non] Confidence: 0.8 Azione[funziona] Confidence: 0.8
S: Hai controllato se il monitor e' connesso correttamente al tuo PC? U: Si, [ho controllato] ed [e' connesso]	Interpretation Confidence: 0.4 Azione[ho controllato] Confidence: 0.2 Azione[e' acceso] Confidence: 0.6
S: Scusa non ho capito, hai detto che il tuo monitor e' connesso?	

U: Sì, esatto	
S: OK, ti passo un operatore che ti possa aiutare	

Table 3 - An example of dialog showing both implicit (turn 2) and explicit confirmation (turn 4)

With reference to Table 3, under the column 'TURNS', dialog turns are reported, system turns are marked with '**S**', user turns are marked with '**U**'. Under the column 'ATTRIBUTE CHUNKING', word chunks, the associated attribute annotation and confidence are reported, together with an overall interpretation confidence, computed as arithmetic mean of chunk level confidence. We show an example of dialog where both implicit and explicit confirmations fire. We use dynamic thresholding, i.e. the confidence threshold used to fire an implicit or explicit confirmation changes at each turn. For the moment the dynamic thresholds are set and tuned by the dialog developer (hard coded), but in the future some experiments will be run in order to find optimal thresholds. There are three confidence thresholds leading the system to four different behaviors along the dialog:

- If the interpretation confidence is under LowerConfidenceThreshold (0.3), the system rejects the interpretation
- If the interpretation confidence is between the LowerConfidenceThreshold (0.3) and ImplicitConfirmationThreshold (0.5), the system perform an explicit confirmation (highlighted in red in Table 10)
- If the interpretation confidence is between the ImplicitConfirmationThreshold (0.5) and UpperConfidenceThreshold (0.7), the system performs an implicit confirmation (highlighted in yellow in Table 10)
- If the interpretation confidence is over UpperConfidenceThreshold (0.7), the interpretation is accepted (highlight is in green in Table 10)

Unlike the previous prototype, this version uses the SLU module, described in Work Package 2, at each dialog turn allowing a pure conversational dialog. In turns like 4, the user is supposed to provide a Yes/No answer, if s/he provides more information, the system has usually a low confidence and so explicit confirmation is needed.

Future work at UT includes dynamic adaptation of thresholds and ontology representation of domain knowledge.

3 Evaluation of a LUNA SLU system with real end-users: the Opinion Mining experiment

The first LUNA spoken dialog prototypes presented in the deliverable D5.3 have not yet been evaluated with real end-users, and therefore no subjective evaluations have been made so far. These experiments are planned in the 3rd year of the project. However some systems using bricks developed through the LUNA project have already been evaluated in a real field test.

In this framework this section presents an evaluation with real end-users of a LUNA SLU system experimented in real conditions at France Telecom. The application targeted is the automatic analysis of spoken messages collected through telephone surveys of FT customers. The goal is to detect highly dissatisfied customers that either ask to be called back or are likely to drop off the service. The end-users of this application are customer representatives that use the "dissatisfaction score" given by the SLU system in order to sort by priority the messages they have to listen to and eventually call back the customers.

The following paragraphs present the context of this study, the corpus used for training the LUNA SLU models and the evaluation carried out with the FT customer representatives.

3.1 Context of this study

Speech mining in recorded conversations is becoming a crucial task for call-center management for quality control and service improvement.

Along the speech mining task, important information that has to be collected is the customers' feedback after using a call-center service. This feedback is usually obtained through a survey performed on a voluntary basis among the customers that have recently called the service. The cost of surveys performed by human operators and the need to collect as many feedbacks as possible have led to the development of automated survey applications where a customer replies to a list of closed questions. The answers to these questions are then analyzed manually or by means of Automatic Speech Recognition (ASR) techniques.

This approach has the advantage to be quite simple to process however there are two major drawbacks: firstly the list of closed questions has to be small to prevent customers dropping off the survey before the end, therefore these questions can't cover all the aspects of a call-center service; secondly the callers are often prone to add comments to their answers to closed questions, being frustrated to have only a limited set of answers not necessarily matching their opinion.

For these reasons an open question like "*Please add any further comments you would like to make on the service*" is often added to the survey. The spoken messages collected thanks to this open question are similar to the *verbatim* transcribed by operators, or collected through WEB-based surveys. However the automatic processing of such messages is particularly challenging as they contain most of the current issues in ASR research: spontaneous non constraint speech, disfluences, telephone speech with channel and background noises.

We presented in a previous study [Camelin N., Bechet F., Damnati G. and De Mori R., 2007] a strategy that consists in the robust detection of subjective opinions about a particular topic in a spoken message. Distributions of positive and negative opinions on several topics were automatically extracted and compared to those obtained with a manual approach. In the evaluation presented here we focus on the **alarm detection** problem in a customer feedback application: we want to characterize each customer's survey with a degree of emergency. All the messages considered as urgent need a quick answer from the call-center service in order to satisfy the customer (and of course prevent him/her dropping off the service!).

This alarm detection strategy has been published in [Camelin N., Damnati G., Bechet F. and De Mori R., 2008].

The proposed strategy is based on a classification scheme that takes into account all the features that can characterize a survey: answers to the closed questions, topics and opinions detected in the open question spoken message, confidence scores from the ASR and Spoken Language Understanding (SLU) modules.

The basic assumption is that the alarm classification process will give a higher score to the surveys containing a lot of redundancy in the expression of the dissatisfaction.

3.2 Telephone survey corpora

In this study we use several survey corpora collected from France Telecom Orange mobile phone customers.

For training the SLU models we use messages transcribed and annotated by human operators according to the following topics: *courtesy* representing the courtesy of the customer service operators, *efficiency* related to the efficiency of the customer service and *rapidity* concerning the amount of time they had to wait on the phone before reaching an operator. Each topic is associated with a positive (+) or negative polarity (-), leading to a set of 6 opinion labels. The topics are not only identified but localised with their associated message segment. This corpus contains about 2400 messages.

The following example illustrates the notion of segments potentially carrying different topics in a same message:

“yes uh uh here is XX XX on the phone well I’ve called the customer service yep <courtesy+> the people were very nice </courtesy+> <efficiency+> I’ve been given valuable information </efficiency+> but <efficiency-> it still doesn’t work </efficiency-> so I still don’t know if I did something wrong [...]”

For the evaluation with end-users a last set of messages has been collected in a field trial that aims to evaluate the alarm detection system presented in this document.

The operators who processed the surveys were asked to rank each customer feedback according to its emergency, an *urgent* message being one left by a customer who needs immediate attention. The documents that were presented to them were processed by our alarm detection system and sorted according to the "dissatisfaction" score given by the system. Therefore the operators could immediately compare the scores given by the system and their own appreciation of the messages.

Four degrees of emergency have been defined for characterizing customers' feedback:

- empty: no understandable speech in the messages left after the open question;
- none: messages that don't require any specific attention (happy customers!);
- moderate: messages expressing the need for a human intervention, but with no urgency;
- urgent: for messages that require an urgent call-back from the service.

This evaluation corpus was collected between January and February 2008: 352 user feedbacks were collected. The proportions of *emergency* labels within this corpus are indicated in following table.

Empty	none	moderate	urgent
7%	20%	23%	50%

Table 4 - Proportions of emergency labels

As we can see, 50% of the messages requires a call-center operator to call-back urgently the customer.

3.3 Alarm detection method

The alarm detection system described in this document is made of a two-step process: firstly the spoken messages collected through the open question prompt are analyzed for extracting all the opinions expressed by the customers thanks to our opinion-mining system; secondly the different opinions extracted, with ASR and SLU confidence measures, as well as the automatically

transcribed answers given by the customers to the closed questions are processed by another classification module in order to estimate the emergency level of each message.

The first step is similar to the concept extraction task of WP2: the opinions of each topic (courtesy, efficiency, rapidity) are considered as WP2 concepts, the polarity (positive or negative) being the value associated to each of them. The amount of handled concepts is lower than for other application supported in the LUNA project but the span of concepts is significantly longer and the level of spontaneity (disfluencies, hesitations...) of spoken messages is much higher. As a result, the overall complexity is comparable.

The concepts are obtained thanks to a segmentation process directly integrated into the ASR system thanks to specific Opinion Language Models. This integration of the ASR and SLU processes are one of the main features pushed by the LUNA project and these specific language models are similar to those used to detect comments in the FT3000 application presented in the first year of the project.

The output of this process is a string of segments, each of them likely to be the support of a given opinion. Each segment is processed by a set of classifiers (decision tree, SVM, Boosting) like those used in WP2. A score taking into account the different decisions taken by the classifiers is associated to the segments for each of the six opinion labels; this corresponds to the probability for the segment to be the support of the opinion label. A threshold is applied to this probability in order to select only the reliable opinion labels for a given segment.

The second step of the process aims to select among all the customers feedback those which require the most urgent human intervention. We made the following assumption: there is a correlation between the negative opinion redundancy in a spoken message and its emergency. In other words, the more a customer is dissatisfied, the more he will express his feelings with negative opinion expressions and the easier it will be for the opinion detection system to detect these expressions. Indeed we used the confidence measures of the opinion detection system as a direct indication of the emergency of a given spoken message. A last classifier, based on a boosting algorithm, takes as input all the previous decisions and confidence scores.

During the classification process, the score output by this last classifier is converted into a confidence score with a logistic function. This score is then normalized to obtain natural values between 0 and 10. We consider that the greater is this score, the more urgent is the message.

These scores are used for ranking the messages for the human operators.

This system is described by Figure 2:

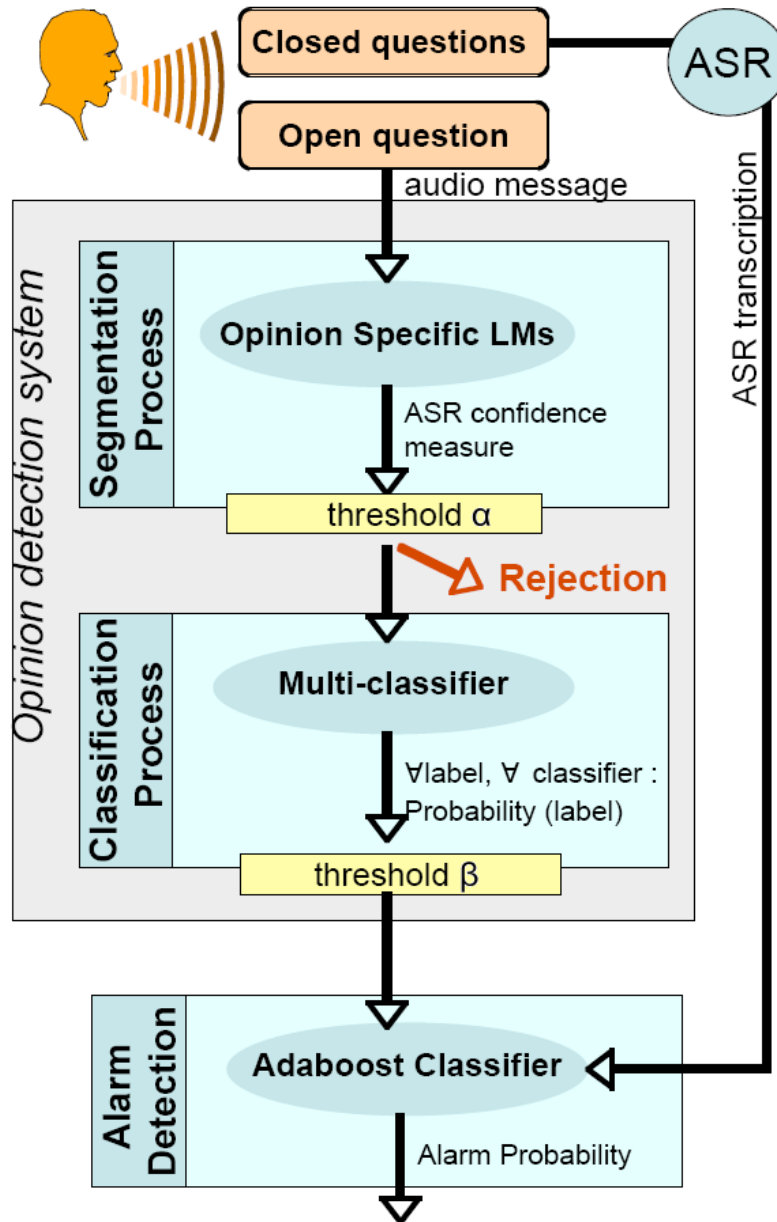


Figure 2 – Alarm detection system

3.4 Evaluation

In order to validate the confidence scores representing the emergency degree of our alarm detection system, we define three levels of emergency according to the score obtained:

- **level1**: scores from 0 to 3 are the first level which represents non-urgent messages
- **level2**: the middle level is made of the scores from 4 to 6
- **level3**: scores from 7 to 10 are the third level which represents the urgent messages

The following table shows the correlation between the emergency labels given by the human annotators and the levels of emergency estimated by our alarm detection system.

(%)	level1	level2	level3
empty	21.4	0.9	0.8
none	33.0	23.4	5.4

moderate	33.1	28.0	7.9
urgent	12.5	47.7	85.9

Table 5 - Correlation between the emergency labels and the levels of emergency estimated by the alarm detection system

There is an obvious correlation between the reference *emergency* labels and the levels obtained automatically by our alarm detection system. Indeed, 85.9% of the messages automatically classified *level3* are considered as *urgent* as opposed to only 12.5% in *level1*.

Furthermore, *level3* contains only 5.4% of messages that do not require a specific attention (*none*). This confirms our assumption that the more users express their dissatisfaction, the easier is the classification task and the higher are the confidence scores obtained.

The repartitions of messages according to the reference *emergency* labels and the confidence scores given by the alarm detection system are shown in the following figure:

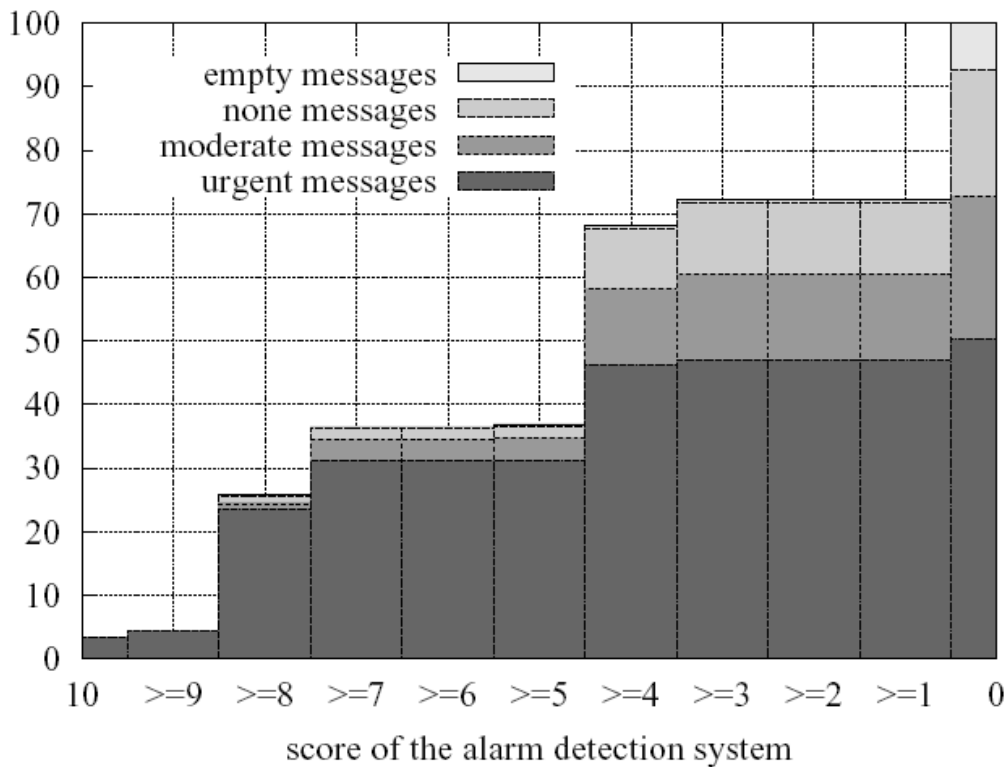


Figure 3 – Repartition of messages

Particularly, it is interesting to analyse the different proportions obtained when selecting all the messages that reach a score upper or equal to 7. At this step, about 36% of the messages in the test corpus are selected and more than 85% of these messages are considered as *urgent* by the human operators. Furthermore, at this step, 62% of the whole *urgent* messages are selected. By comparison, considering only answers to closed questions and the same corpus coverage, only 76% of *urgent* messages are selected, representing only 55% of the whole *urgent* messages.

This field experiment carried out at France Telecom has validated our assumption that dissatisfied customers express more clearly their negative opinions, leading the opinion detection system to produce better confidence scores that enables the alarm detection system to finely analyse the emergency degree of a message.

Moreover we had very positive subjective evaluation from the customer representatives using the system: before they used a random selection of the customer messages they had to listen to, and therefore they had to listen to a lot of empty or not informative messages (27% of the messages). With our system the risk of having such messages drops to only 6%, and the feedback we have from them was that even if the alarm detection score was not always accurate, this filtering and selection process was very useful optimizing the amount of "interesting" messages that could be handled during a listening session.

4 Other results

4.1 Imposing semantic coherence

Let $\{F_d\}$ be a set of frames introduced for describing the semantic knowledge of an application domain. A frame F_d has a name and roles represented by slots $\{\gamma_{d,j}\}$. For the sake of simplicity, a frame will be indicated with its name. A frame instance is a *semantic structure* characterized by a set of slot values. A slot is called a *property* when its value is the instance of another frame, otherwise it is called an *attribute*.

For example there is frame RESERVATION has the following structure:

```
RESERVATION    {
                is_a NEGOTIATION_ITEM
                attributes
                  status
                  quantity
                  file_numbe
                properties
                  customer [PERSON]
                  res_time [TIME]
                  theme [LODGING, RESTAURANT]
                  cost [PRICE]
                }
```

A semantic relation $R_{d,j}$ exists between a frame F_d and each of its slots $\gamma_{d,j}$. The fact that a semantic relation is coded into a sentence is supported by patterns π_{djk} made of sentence linguistic features like words.

A *frame instance fragment* $\Gamma_{i,a}$ is made of a frame name and a slot list represented as:

$$\Gamma_{i,a} = F_i .sl_{i,a}$$

A slot list $.sl_{i,a}$ is represented as follows as a set of slot names and slot values:

$$.sl_{i,a} = s_{i,a,1} \cdot v_{i,a,1}, \dots, s_{i,a,k} \cdot v_{i,a,k}, \dots, s_{i,a,K} \cdot v_{i,a,K}$$

where $s_{i,a,k}$ is a slot name and $v_{i,a,k}$ is a slot value of a slot $\gamma_{i,a,k}$. Some slot values can be represented by variables to be bound by an interpretation process. Each slot $\gamma_{i,a,k}$ is associated

a *facet* $\Phi_{i,a,k}$ which is a list of value types. Value types can be names of frames whose instance may be a possible value. A facet is represented as follows:

$$\Phi_{i,a,k} = \{\varphi_{i,a,k,j}\}$$

$$\varphi_{i,a,k,j} = F_q \quad \text{or} \quad V_q$$

F_q is a frame name while V_q is another type suitable for an attribute values.

Frame F_i is the *head* of the frame instance fragment. All the unfilled slots of the instance fragment are elements of the *fragment tail* list.

An instance of this frame may contain values only for some attributes and properties represented as follows:

RESERVATION

status requested
theme instance of ROOM
 room_type *single_bed*

Complex semantic structures, consistent with the application ontology, can be obtained by composition with already hypothesized semantic structures if there is a syntactic support in the data to be interpreted.

Possible compositions are obtained by matching the head of a structure with the content of the facets associated with the tails of another structure. Let assume that two instances $\Gamma_{i,a}$ and $\Gamma_{h,b}$ have been hypothesized and that a syntactic relation $\text{SYNR}_{i,h}$ has been found to exist between them. The instances can be composed into a new structure $\Gamma_{i,c} = \Gamma_{i,a} \cdot \Gamma_{h,b}$ if the head F_h appears in the facet of a tail of $\Gamma_{i,a}$. Another possible composition is $\Gamma_{h,c} = \Gamma_{h,b} \cdot \Gamma_{i,a}$ if the head F_i appears in the facet of the tail of $\Gamma_{h,b}$. A possible composition compatible with the application ontology is represented as follows.

$$\Gamma_{i,h} = R_j(\Gamma_{i,h}) \cdot \tau_{i,a,j}(\Gamma_{i,a}) \cdot F_h \cdot sI_{h,b}$$

where $\tau_{i,a,j}(\Gamma_{i,a})$ is the j-th tail of $\Gamma_{i,a}$, $R_j(\Gamma_{i,h})$ is the fragment obtained from $\Gamma_{i,a}$ by removing the tail $\tau_{i,a,j}(\Gamma_{i,a})$.

In the above example, the instance fragment has head RESERVATION and the tail list contains $\{customer, res_time, theme, cost\}$ and the unfilled slots of the frame ROOM. The facet of *cost* contains the frame name PRICE. Thus a fragment with head PRICE can be composed with the RESERVATION fragment as filler for the slot *cost*.

A semi automatic procedure was used for annotating the manual transcriptions of the entire MEDIA corpus. A set of 10000 turns was manually corrected/validated. In this set there are 2248 turns belonging to the MEDIA development and test set. This set will be indicated as *set2* and was used for evaluating compositions and for using semantic constraints for validating or correcting the output of the confidence CRF, introduced in deliverable D2.3.

Just based on the confidence CRF results on the one-best ASR output; the same structure as the one provided by manual annotation. It was hypothesized for 76.4% of the dialogue turns. It raised

to 79.8% with the simple correction procedures outlined below. Correction procedures are applied only in situations characterized by preconditions. Preconditions are characterized by some never observed co-occurrence of constituents or by the absence of essential frequently observed components in specific contexts.

1. Possible deletions of speech acts were hypothesized based on the absence of relative constituents in specific contexts of other constituents. For example, if only an instance of PRICE is detected in a speech turn, the application ontology suggests that there may be a QUERY act. The support for this speech act may have been not detected because it is made in French by a short sequence of monosyllabic words, one of which has been deleted by the ASR system. If a suitable support is found for this hypothesis is found in the word graph with sufficient evidence, then the speech act is inserted. References expressed by the subject of a verb can be inferred by the verb itself and asserted because in many French syntactic structures, verbs cannot be omitted.
2. Possible insertions are due to hesitations, repetition or corrections of sequences of numbers. In this case an instance of NUMBER has to be deleted. If a support for this action is found in the word graph, then the hypothesis is deleted.
3. Possible substitutions depend on the fact that a constituent is specified by the context of other constituents. For example a request of a number of rooms turns an instance of NUMBER into an instance of *room_availability*.
4. Other errors are due to homophones whose transcription can be corrected based on predicted semantic constituents and syntactic agreement.

The CER results on set2 are reported in Table 6. The oracle CER is 19.9.

CRF	FST	SVM	POS	conf CRF	Sem corr
22.2	27.4	25.4	24.9	21.3	20.1

Table 6 – Percentage CER on the ASR outputs of set2

The results in Table 4 show that the attempt to impose semantic constraints on the formulation of constituent hypotheses is promising.

4.2 Frame annotation in Polish

Preliminary tests were performed concerning the predicate level of human-human dialogs annotation.

The linguistic description of verb frames is based on the Berkeley FrameNet (with small modifications) and at first concerns verbs of movement that are of great importance in the Polish LUNA dialog domain. For the experiment, a list of 52 verbs were selected, whose description contains 776 frame realizations.

A program was written which uses earlier prepared verb description to search the corpus for defined frames and to assign to each verb a frame and a set of frame elements. The information of turns borders and attributes is used and described at the domain level. It can happen that the program gives more than one frame interpretation for the same verb. The selection among these candidates is done manually during post-annotation verification.

The test runs on randomly chosen 15 files from the corpus. These files contain 76 verbs of different types, in particular 31 verbs of movement.

An example of the annotation is presented below.

The example of sentence

o której | odjeżdża | ostatni autobus | z przystanku Powstańców | w stronę
what time does the last bus go from bus-stop Powstancow in direction
TIME | DEPARTING | THEME | SOURCE | GOAL
Wiatracznej | sto czterdzieści pięć numer
Wiatraczna one hundred forty five number
| THEME

```
<Frames>
<Set id='1' subset='1' span='word_14' frame='DEPARTING' head='odjeżdżać'>
<Frame fe='1' span='word_12..word_15' slot='TIME' />
<Frame fe='2' span='word_16' slot='THEME' />
<Frame fe='3' span='word_17..word_19' slot='SOURCE' />
<Frame fe='4' span='word_20..word_22' slot='GOAL' />
<Frame fe='5' span='word_23..word_26' slot='THEME' />
</Set>
</Frames>
```

5 References

Moschitti A., Riccardi G. and Raymond C., **Spoken Language Understanding with Kernels for Syntactic/Semantic Structures**, Proc. IEEE ASRU Workshop, Kyoto, 2007.

Varges S. Riccardi G. and Quarteroni S., **Persistent Information State in a Data-Centric Architecture**, SIGdial Workshop on Discourse and Dialogue, Columbus, 2008.

Camelin N., Bechet F., Damnati G. and De Mori R., **Speech mining in noisy audio message corpus**. In Proceedings of InterSpeech, Antwerp, Belgium, September 2007.

Camelin N., Damnati G., Bechet F. and De Mori R., **Automatic customer feedback processing: alarm detection in open question spoken messages** In Proceedings of InterSpeech, Brisbane, Australia, September 2008.