

CLASSiC

D5.5: Advanced Appointment-Scheduling System “System 4”

Romain Laroche, Ghislain Putois

Distribution: Consortium

CLASSiC

Computational Learning in Adaptive Systems for Spoken Conversation
216594 Deliverable 5.5

December 2010



Project funded by the European Community
under the Seventh Framework Programme for
Research and Technological Development



The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	216594
Project acronym	CLASSiC
Project full title	Computational Learning in Adaptive Systems for Spoken Conversation
Instrument	STREP
Thematic Priority	Cognitive Systems, Interaction, and Robotics
Start date / duration	01 March 2008 / 36 Months

Security	Consortium
Contractual date of delivery	M33 = December 2010
Actual date of delivery	December 2010
Deliverable number	5.5
Deliverable title	D5.5: Advanced Appointment-Scheduling System “System 4”
Type	Prototype
Status & version	final 1.0
Number of pages	15 (excluding front matter)
Contributing WP	5
WP/Task responsible	WP5, leader France Telecom
Other contributors	
Author(s)	Romain Laroche, Ghislain Putois
EC Project Officer	Philippe Gelin
Keywords	System, uncertainty management, Dialogue Management

The partners in CLASSiC are:

Heriot-Watt University	HWU
University of Cambridge	UCAM
University of Geneva	GENE
Ecole Supérieure d'Electricité	SUPELEC
France Telecom/ Orange Labs	FT
University of Edinburgh HCRC	EDIN

For copies of reports, updates on project activities and other CLASSiC-related information, contact:

The CLASSiC Project Co-ordinator:
Dr. Oliver Lemon
School of Mathematical and Computer Sciences (MACS)
Heriot-Watt University
Edinburgh
EH14 4AS
United Kingdom
O.Lemon@hw.ac.uk
Phone +44 (131) 451 3782 - Fax +44 (0)131 451 3327

Copies of reports and other material can also be accessed via the project's administration homepage,
<http://www.classic-project.org>

©2010, The Individual Authors.

No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.

Contents

Executive Summary	1
1. Introduction	2
2. Service organisation	2
2.1. Complete dialogue examples	2
2.2. Dialogue strategies	3
2.3. ASR reject recovery strategies	4
3. System organisation	5
3.1. Automatic Speech Recognition (ASR)	5
3.2. Spoken language understanding (SLU)	6
3.3. Dialogue management (DM)	6
3.4. Natural language generation (NLG)	6
3.5. Text-to-speech (TTS)	6
4. System experimentation	6
A. ASR <i>N</i>-best list calibration and combination	7
A.1. Introduction	7
A.2. Study Environment	7
A.2.1. French ASR off-the-shelf solution	7
A.2.2. The corpus and the language model	8
A.2.3. Evaluation metrics	8
A.3. Confidence measure calibration	9
A.3.1. Calibration methods	9
A.3.2. Issues with ICE evaluation	9
A.3.3. ICE evaluation after calibration	10
A.4. Probabilistic combination of several ASR hypotheses	11
A.4.1. Transformation of probabilities	11
A.4.2. ICE evaluation after combination	12
A.5. Next steps	13

Executive Summary

This document is a short report to accompany the Prototype Deliverable D5.5, due at month 33 of the CLASSIC project. It describes an extended Appointment Scheduling service and the CLASSIC System 4 supporting it. First an overview of the service organisation is given, then the implementation of the main changes since the CLASSIC System 3 is illustrated.

1. Introduction

This document describes the laboratory Appointment Scheduling service built on an extension of the industrial final CLASSiC System 3 described in deliverable D5.3.4 [1]. The application domain selected is how to make an appointment with a technician for an intervention at home, when one's landline is down. The target language is French only. CLASSiC System 4 is a laboratory version that aims at being a more flexible system than the CLASSiC System 3.

Section 2 presents the service organisation. And section 3 gives an overview of the evolutions that were implemented in CLASSiC System 4 from CLASSiC System 3.

2. Service organisation

The service is intended to help a novice user obtain quickly an appointment with a technician when her landline is down. The part improved in the CLASSiC project is inserted inside a larger automaton-based troubleshooting system, which first identifies the caller and the reason for her call, then performs some automated line tests, before running the Appointment Scheduling System 4.

2.1. Complete dialogue examples

Here are two complete sample dialogue sessions made with the system. The first one is very simple.

- SYSTEM : Bonjour et bienvenue. La réparation de votre ligne nécessite l'intervention d'un technicien. Quand souhaitez-vous prendre rendez-vous ?
- USER : Cette semaine.
- S : Vous avez une préférence pour cette semaine. Merci de préciser le jour de la semaine pour votre rendez-vous. Par exemple : mercredi.
- U : Ce mercredi, plutôt dans la matinée.
- S : Je vous prie de bien vouloir m'excuser. Je ne suis pas sûr d'avoir bien compris. Avez-vous dit : mercredi ? Merci de confirmer par oui ou par non.
- U : oui.
- S : Merci de confirmer la réservation du créneau suivant. Le mercredi 14 juillet avec le passage du technicien dans la matinée. Est-ce que cela vous convient ?
- U : oui.
- S : Votre rendez-vous a bien été enregistré. Merci d'avoir utilisé notre service. Au revoir.

In this second example, the user prompts are associated with the confidence score computed by Telisma. It is far more complex since the user and the system need several attempts to find a matching appointment.

- SYSTEM : Bonjour et bienvenue [pause 1sec]. La réparation de votre ligne nécessite l'intervention d'un technicien. Quand souhaitez-vous prendre rendez-vous ?
- USER : jeudi après-midi < 74 >
- S : Merci de confirmer la réservation du créneau suivant : le Jeudi 22 juillet avec le passage du technicien dans l'après-midi. Est-ce que cela vous convient ?
- U : non < 75 >

- S : Vous venez de refuser un créneau. Aucun autre créneau ne satisfait vos contraintes. Nous allons reprendre depuis le début. Essayez de spécifier d'autres contraintes. Quand souhaitez-vous prendre rendez-vous ?
- U : samedi matin < 70 >
- S : Vous avez une préférence pour Samedi dans la matinée. Je suis désolé. Nous n'avons trouvé aucun créneau satisfaisant vos contraintes. Nous allons reprendre depuis le début. Essayez de spécifier d'autres contraintes. Quand souhaitez-vous prendre rendez-vous ?
- U : le lundi matin < 60 >
- S : Désolé, je ne suis pas sûr d'avoir bien compris. Avez-vous dit Lundi dans la matinée ? Merci de confirmer par oui ou par non.
- U : oui < 72 >
- S : Vous avez une préférence pour Lundi dans la matinée. Je suis désolé. Nous n'avons trouvé aucun créneau satisfaisant vos contraintes. Nous allons reprendre depuis le début. Essayez de spécifier d'autres contraintes. Quand souhaitez-vous prendre rendez-vous ?
- U : jeudi matin < 70 >
- S : Vous avez une préférence pour Jeudi dans la matinée. Je suis désolé. Nous n'avons trouvé aucun créneau satisfaisant vos contraintes. Nous allons reprendre depuis le début. Essayez de spécifier d'autres contraintes. Quand souhaitez-vous prendre rendez-vous ?
- U : le vendredi le quinze < 69 >
- S : Désolé, je ne suis pas sûr d'avoir bien compris. Je répète. Quand souhaitez-vous prendre rendez-vous ?
- U : ce vendredi matin < 67 >
- S : Désolé, je ne suis pas sûr d'avoir bien compris. Avez-vous dit : Vendredi de cette semaine dans la matinée. Merci de confirmer par oui ou par non.
- U : oui < 82 >
- S : Vous avez une préférence pour Vendredi de cette semaine dans la matinée. Je suis désolé. Nous n'avons trouvé aucun créneau satisfaisant vos contraintes. Nous allons reprendre depuis le début. Essayez de spécifier d'autres contraintes. Quand souhaitez-vous prendre rendez-vous ?
- U : vendredi après-midi < 72 >
- S : Vous avez une préférence pour Vendredi dans l'après-midi. Souhaitez-vous un rendez-vous pour cette semaine ou la semaine prochaine ?
- U : cette semaine < 75 >
- S : Merci de confirmer la réservation du créneau suivant. Le Vendredi 16 juillet avec le passage du technicien dans l'après-midi. Est-ce que cela vous convient ?
- U : oui < 79 >
- S : Votre rendez-vous a bien été enregistré. Merci d'avoir utilisé notre service. Au revoir.

2.2. Dialogue strategies

The Appointment Scheduling is organised in rounds of negotiations. In each round, the system attempts to find a new timeslot suitable for the user. Two main strategies are used by the system : either the system directly proposes a timeslot to the user (system initiative strategy), or it asks the user for several pieces

of information about either a day, a week or a period during the week she would be available, until being able to propose a timeslot matching the user's constraints (user's strategy).

CLASSiC System 3's online learning [2, 3, 4] showed that the best plan is to first try with system's initiative strategy and then try the user's initiative strategy. However, CLASSiC System 3 was only capable to understand a very formatted and complete user's initiative utterance, such as "mercredi après-midi".

CLASSiC System 4 was built to accept various ("demain", "cette semaine", "le douze"¹ kinds of entries in the user's initiative strategy. In order to be able to study it more deeply, we chose for CLASSiC System 4 to favour the user's initiative strategy and to switch to system's initiative only after two kinds of events : as a recovery strategy after an ASR reject (see section 2.3) or when the user asks for an appointment "dès que possible"².

In order to help the user to know which slot needs to be informed, the dialogue management selects the most discriminant criterion given the system agenda and the current user's constraints thanks to a very simple entropy calculation maximisation.

2.3. ASR reject recovery strategies

Sometimes, ASR returns a reject for an audio input. It either means that the confidence level for the best transcription is too low or that nothing in the language model has fit the input. Such rejects are common and we have experienced in CLASSiC System 4 five alternatives to deal with such errors :

- Feedback + repetition : the system informs the user that it is not sure of what it has heard and then it repeats exactly the same question. This is somehow the baseline for the study, as implemented in CLASSiC System 3.
- Feedback + energetic prompt : as well, the system feedbacks the reject but instead of repeating the same .wav file, we use a more energetic TTS variant that stresses important parts of the question.
- Feedback + rephrasing : the system still feedbacks but rephrases instead the question in another way.
- Feedback + yes/no question : if the ASR's 1-best transcription is not too low, the system asks the question : "Avez-vous dit *1-best* ? Merci de confirmer par oui ou par non."³
- Feedback + change of strategy : the system still feedbacks the reject, but it switches to a system initiative strategy (as described in the previous subsection).

CLASSiC System 4 uses the architecture described in deliverable d5.1.2 [5] but where its policy is not updated in order to keep a system that is constant during the whole experimentation. Thus, the choice between these recovery strategies is random.

Each user's n-best answer is submitted to a confidence test based on the ASR acoustics score, and its uncertainty is integrated into the context manager. For the confirmation question, the distribution of answers output by the ASR module are used to shift the ASR confidence level score of the 1-best and give better precision and recall performances.

¹"tomorrow", "this week", "the twelfth"

²"as soon as possible"

³"Did you say *1-best* ? Please answer with yes or no."

3. System organisation

The System 4 conforms to the CLASSiC architecture specified in deliverable D5.1.2 : it consists of 5 distinct modules: automatic speech recognition (ASR), semantic decoding or spoken language understanding (SLU), dialogue management (DM), natural language generation (NLG) and speech synthesis or text-to-speech (TTS). Each module is built on an industrial component from France Telecom, except for the ASR module which is provided by the third-party vendor Telisma.

The system is built on the France Telecom VXML platform : a vocal platform (Orange Media Server) and an application server (Disserto runtime). Orange Media Server is a VXML telephony platform, which embeds the speech recogniser engine Telisma Telispeech and the text-to-speech engine France Telecom Baratinoo. Disserto is both the application server and the dialogue design suite. It is composed of two runtime components (phase engine, semantic analyser) and design tools (Dialogue Design Studio, Dialogue Analyser Studio). The runtime components are deployed in an Apache Tomcat servlet container.

3.1. Automatic Speech Recognition (ASR)

The ASR module uses the Telisma commercial Telispeech speech recogniser. It gets live speech input from a dedicated telephony board and outputs N-best hypotheses with sentence-level confidence scores.

While CLASSiC Systems 3 and 4 have the same grammar for interpreting the “yes/no” answers, CLASSiC System 3 allows only very formatted utterances from the user for the question “Quand êtes-vous disponible ?”⁴. Indeed, only complete queries DAY_OF_THE_WEEK/PERIOD_OF_THE_DAY such as “mercredi matin”⁵ are accepted. CLASSiC System 4 is much more flexible and is supposed to accept any french constraints except disjunctions of this kind: “mardi ou mercredi”⁶ and negative constraints such as: “pas lundi”⁷. We made this choice for two reasons. First, these kinds of constraints are not natural. In our experience, people never use this kind of complex expressions. And secondly, it has always been difficult to code logical expressions in conjunction with probabilistic representation.

In this final version of System 4, the dictionary is about 200 words: numbers (“le premier”, “le dix”, ...), months (“juillet”, “août”, ...), day of the week (“lundi”, “mardi”, ...), relative expressions for days/weeks (“ce jeudi”, “demain”, “cette semaine”, ...), yes and no (“absolument”, “pas du tout”, ...), request for repetition (“pardon, je n’ai pas compris”, “répète”, ...), request for an initiative switch (“SYSTEM : Quand êtes vous disponible ?”, “USER : Dès que possible.”), All combinations of constraints are possible. It is for instance possible to understand “la semaine prochaine dans la matinée”⁸.

As Telispeech’s ASR confidence measure does not deliver probabilities, we had to calibrate and combine the ASR results. The calibration process consists of observing the confidence scores on an annotated corpus and to calibrate the probability of a score being true to the probability that was observed on this corpus. The combination process consists of considering each ASR result in the n-best list as an independent probabilised piece of information and to combine their probabilities in order to compute a probability distribution on the n-best list (see appendix A).

⁴“When are you available ?”

⁵“Wednesday morning”

⁶Tuesday or Wednesday

⁷not on Monday

⁸next week in the morning

3.2. Spoken language understanding (SLU)

The SLU module is called the semantic analyser. It is a part of the Disserto suite. The semantic analyser in this service handles around 50 concepts. It uses two parsing stages. The first stage is a tagger, which associates morphological forms with tags. The second stage is a rule extractor, which associates sequences of tags with a resulting parametrised rule. The tag and rules files are designed with the Dialogue Analyser Studio, the dedicated Disserto SLU design tool.

The SLU implementation of CLASSiC System 4 is far more complex than CLASSiC System 3's. Indeed, there are a lot of possible combinations between the constraints expressions.

3.3. Dialogue management (DM)

The DM module is called the phase engine. It is also a part of the Disserto suite. The phase engine is a JAVA servlet which executes the service automaton. The service automaton is handcrafted beforehand with the Dialogue Design Studio, a major component of the Disserto suite.

CLASSiC Systems 3 and 4 have DM implementations that are completely different. Indeed, there are deep differences in dialogue logics that have been evoked in sections 2.2 and 2.3.

3.4. Natural language generation (NLG)

The current system does not use a NLG component. Instead, it relies on handcrafted prompts manually designed in the Dialogue Design Studio.

The prompts have been specially designed for CLASSiC System 4. Once more, nothing was recycled from CLASSiC System 3.

3.5. Text-to-speech (TTS)

The TTS component synthesises speech from handcrafted prompts using the Baratinoo expressive speech synthesiser. Baratinoo is an industrial speech synthesiser using state-of-the-art unit selection technology, especially tuned for French synthesis (voice Loïc). Baratinoo is compliant to all current standards for text input (SSML), pronunciation specification (PLS) and APIs (SAPI, MRCP). Very schematically, Baratinoo operates in two steps: a linguistic analysis step that converts the input text to a sequence of phonemes, and a unit selection step that generates the speech signal from the sequence of phonemes. The unit selection step consists of finding the best sequence of acoustic units in the recorded speech database with a Viterbi search, and concatenating them.

The same voice (Julie) was used for CLASSiC Systems 3 and 4. But, as the prompts are different, the whole unit selection step had to be reengineered.

4. System experimentation

The system has been publicly deployed for testing since November 2010. It has now collected 887 calls.

A. ASR N -best list calibration and combination

A.1. Introduction

Last decade, Confidence Measures (CM) for Automatic Speech Recognition (ASR) have been the subject of a lot of effort in the scientific literature. There are three major categories for CM [6]: CM as combination of predictor features [7], CM as posterior probability [8] and CM as utterance verification [9]. All of them aim at providing the speech applications with a probability as close as possible from the observed probability, *i.e.* 80% of recognition results with CM equals to 0.8 should be correctly recognised.

Since a long time, ASR off-the-shelf solutions provide the system with a measured N -best list of utterances. But, these CM are obtained in a very obscure way and they generally present two main flaws. First, they are far from being reliable as observed probability predictions. Second, each of their measures are generated separately, so that the sum of probabilities over the possible utterances are not guaranteed to be under 1.

Two of the aforementioned CM categories require information that are inaccessible in off-the-shelf ASR (predictor features and confusion network of posterior probabilities). The third one (utterance verification) was very popular around year 2000 but got forgotten while posterior probability became more and more influential. The principle of utterance verification is to apply a post-processing to ASR output in order to assign CM. The idea defended in this paper is quite close : we propose to post-process CM obtained with the off-the-shelf ASR in order to make them fit the observed probabilities.

Our method aims at exploiting ASR CM into commercial dialogue applications with uncertainty management [10, 11]. This paper presents a straightforward method to improve CM, first by confronting the predicted probabilities with the observed ones. It differs from utterance verification because it applies a CM calibration instead of computing one new from scratch. Another difference is that, afterwards, we combine each calibrated CM in the N -best list in order to deliver a probabilistic distribution.

Section A.2 presents the ASR solution, the corpus and the evaluation metrics that have been used during experiments. Section A.3 proposes different calibrations. Section A.4 applies a mathematically sound probability combination on the calibrated CM. Finally, section A.5 concludes the paper with next steps.

A.2. Study Environment

A.2.1. French ASR off-the-shelf solution

We tried to lead our study with the two main ASR solutions existing in french: Nuance's and Telisma's. It appeared that Nuance was providing very rarely several elements in the N -best (even with a maximal speedVSaccuracy parameter), and that the CM were given as probabilities with a very small preciseness (only integer percentages and with very few values between 10% and 90%). We highly suspect the Nuance software to already post-process the CM provided by their confusion network in order to propose actual probabilities. As a consequence, Nuance's ASR does not deliver all the information it has in first hand but only what remains after their post-processing. On the contrary, Telisma provides raw and complete N -best lists with many-valuated CM between 0 and 1000 that are supposed to interpretable as probabilities once divided by 1000.

Thus, during the whole paper, experiments are made with Telisma's ASR solution and ASR CM refer to the ones delivered by Telisma's ASR solution. The full N -best list is processed and the speedVSaccuracy parameter has been set to 43 on a scale of 100 (this is the value we use for commercial applications).

A.2.2. The corpus and the language model

In this paper, our experiments are restricted to the very simple yes/no language model (in french), so that there are only 3 possible classifications for each user's utterance: "yes", "no" and "repeat". Synonyms such as "that's it", "absolutely", "not in the least" and others are also recognised and classified accordingly in the yes-no-repeat classes. This has been done in order to take benefit of bigger corpora and to provide the reader with results easy to interpret.

We have three sets of corpus (MPP2, MPP3-1 and MPP3-2) of approximately equal size. All of them were acquired with real customers calling a different release of a commercial application on landline troubleshooting. A fourth corpus has been gathered from a business telephone directory application (PhoDir). This corpus only gathered actual yes/no answers (and no reject), which explains the very low error rate and the fact that we will only use it for evaluation (and not for learning purpose). In many respects, this application is different from the other one: first, the line troubleshooting application receives a lot of mobile calls when the telephone directory receives exclusively office phone calls and second the former is rarely used several times by a single person when the latter was used repeatedly by the same persons. The volumes and percentage of the 1-best error rate of Telisma are provided in the following table.

Table 1: Yes/no corpora features.

Corpus	MPP2	MPP3-1	MPP3-2	PhoDir
Volume	2812	3345	3877	27879
Teli's error rate	13.84%	8.37%	6.24%	1.40%

The high percentage of error rates are partially due to the fact that people are often not answering the yes/no question but either are repeating what the system said for confirmation (when the system asks "are you at home?", the user answers "I'm at home" instead of "yes"), are talking in an aside (talks to someone else in the room: "turn down the TV") or are complaining about their personal problems (off-topic: "because, you know, my grand-daughter touched the phone..."). This kind of user behaviour represents almost 12% of the MPP2 corpus and is barely existent in the PhoDir one. Thus, the high-variability among the error rates is explained by the fact that the application releases are different and the system questions are probably generated in a better way, so that the user keeps more focussed and uses the proper "yes" or "no" answers.

A.2.3. Evaluation metrics

For the evaluation, we use the Item-level Cross Entropy [12] (ICE). This metrics gives a way to evaluate CM and the overall correctness. It's a cross entropy between the probability density from the CM and the optimal density given by the correct value. As we work at the utterance classification level and not at the item-in-utterance level, we adapt it in a straightforward way. The mathematical definition is given below:

$$ICE = \frac{1}{N} \sum_{u,i} -\log(\delta_{uC_i} P_{uC_i} + (1 - \delta_{uC_i})(1 - P_{uC_i})) \quad (1)$$

Where u stands for the user utterance, C_i is a recognized class, N , the number of utterances in the corpus, δ_{uC_i} equals 1 if utterance u belongs to class C_i and 0 otherwise and P_{uC_i} is the CM (expressed as a probability) affected to C_i given utterance u .

A.3. Confidence measure calibration

The goal of this section is to calibrate ASR CM with observed probabilities. The baseline is this measure divided by 1000 (thus between 0 and 1).

A.3.1. Calibration methods

First, a very simple k nearest neighbours (k NN) algorithm is applied: for an ASR CM P , the algorithm looks for the k nearest CM in the corpus and computes the correctness rate over this set. This correctness rate is the calibration of P . In our testing, $k = 200$ seemed to be generalising enough to ensure an almost monotone curve.

Figure 1 shows that k NN generalisation were very close for all three learning corpus. It also reveals that the sigmoid function is a very good trend curve to model the calibration:

$$S_{\alpha\beta\gamma}(x) = \gamma + \frac{1 - \gamma}{1 + e^{-\alpha(x-\beta)}} \quad (2)$$

Where α and β respectively define the derivative value and the abscissa at inflection point. The only area of the graphic where k NN curves does not fit a classical sigmoid function is for low probabilities. This is the reason why we added the γ parameter which approximately⁹ defines the value at abscissa equal to 0. We explain this by the fact that number of classes is low (only 3) and that even with a low confidence on a recognition result, there is still a significant probability that this result is right by “chance”. The sigmoid parameters are optimised with a simple gradient descent algorithm in order to minimise the ICE. Previous works on CM already identified the sigmoid [13].

A.3.2. Issues with ICE evaluation

There are some issues for ICE evaluation. First, the ASR module does not always find a correspondence for each class, meaning that some class-utterance couples may have a zero-probability although the utterance belongs to this class. This results in an infinite ICE. Calibration solves the first ICE evaluation issue since $S_{\alpha\beta\gamma}(x) > \gamma \geq 0$ and since the k NN always returns a non 0 value with $k = 200$. In order to get comparable ICE, we needed to artificially fix this problem for the ASR CM: we affect class-utterance tuples with a minimum γ^* probability that we determine during the sigmoid optimisation. Thus, the ASR CM is updated as follows:

$$P_{uC} = \max(\gamma^*, P_{uC}) \quad (3)$$

Second, some user utterance cannot be interpreted as belonging to any of the three classes. This is the reason why we added a fourth class: “out-of-vocabulary” (C_{oov}). But the sum of CM we obtain so far are not guaranteed to be under 1. Thus, we need to define a probability of reject as follows:

$$P_{uC_{oov}} = \max\left(0.01, 1 - \sum_i P_{uC_i}\right) \quad (4)$$

This second issue is fixed thanks to probability combination (see section A.4).

⁹Actually it defines the limit at $-\infty$.

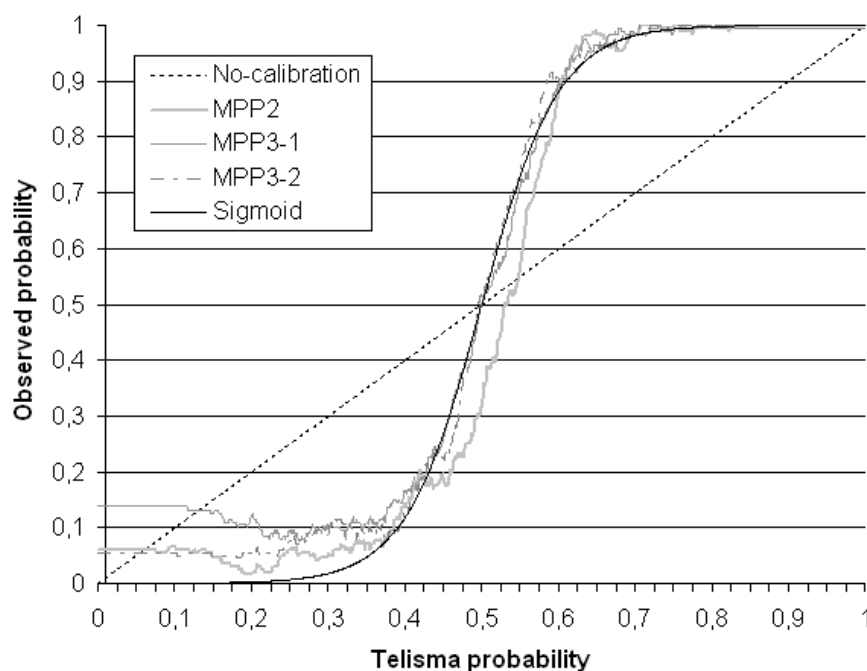


Figure 1: $k = 200$ nearest neighbours calibration for each learning corpus and the sigmoid function.

A.3.3. ICE evaluation after calibration

The following table shows the cross learning/evaluation where each corpus is successively used as the learning corpus and the test corpus. For reference, the ICE obtained with ASR CM on each corpus are also provided.

Calibr.	ASR	k NN			Sigmoid		
		MPP2	MPP3-1	MPP3-2	MPP2	MPP3-1	MPP3-2
MPP2	0.92	0.45	0.46	0.46	0.46	0.49	0.48
MPP3-1	0.91	0.57	0.51	0.56	0.53	0.51	0.51
MPP3-2	0.86	0.47	0.42	0.45	0.44	0.42	0.42
PhoDir	0.62	0.25	0.21	0.23	0.22	0.19	0.20

This table clearly shows that the calibration has a strong effect on ICE. Both the k NN algorithm and the sigmoid optimisation approximately divide ICE by 2. Clearly, ASR CM holds very interesting probabilistic information once it has been calibrated. The average ICE performance on off-learning evaluations¹⁰ is 0.40 for k NN and 0.39 for the sigmoid optimisation, from a 0.83 average ASR performance.

The optimal values for the sigmoid function are approximately (slightly depends on the learning corpus): $\alpha = 20$, $\beta = 0.5$ and $\gamma = 0.1$. The learnt sigmoids for MPP3-1 and MPP3-2 are very close. For MPP2, the values sensibly differ, but in the end, sigmoid-calibrated performance remains far better than the ASR's.

¹⁰When the learning corpus (columns) is not the same as the tested corpus (rows).

A.4. Probabilistic combination of several ASR hypotheses

Now that probabilities are calibrated, we are still facing the second hinder described during the introduction: the probabilities of the N -best list are obtained independently from each other, so that it frequently happens that the sum over the N -best list exceeds 1. The probabilities we have processed through calibration in section A.3 only inform the probability that the input signal matches each utterance independently from each other. But it does not provide the probabilities that the input signal in fact delivers each utterance. This section provides the probabilistic tools for such a transformation.

A.4.1. Transformation of probabilities

In this subsection, we suppose that N -best list calibrated probabilities are independent. This is acceptable for yes/no question and in general for ASR classification. However, it is a strong restriction about open speech recognition.

$$ASRresult = \{C_k, P_{uC_k}\}_{k \in [0, n-1]} \quad (5)$$

Where P_{uC_k} is the calibrated probability (if any calibration is applied) that utterance u matches class C_k . The probability $P_{uC_k}^*$ that the u belongs to class C_k is the probability that C_k matches u and that for all $i \neq k$, C_i does not match u . We compute this probability as follow:

$$P_{uC_k}^* = \frac{P_{uC_k}}{\kappa} \prod_{i \neq k} (1 - P_{uC_i}) \quad (6)$$

Where κ is a normalising constant. κ is computed by summing all $P_{uC_k}^*$ and the probability $P_{uC_{oov}}^*$ that u does not match any class (out of vocabulary). The probability that the input is out of vocabulary is the joint probability that it does not match any class C_i (equation 7). We also know that all these probabilities sum to 1 (equation 8).

$$P_{uC_{oov}}^* = \frac{1}{\kappa} \prod_i (1 - P_{uC_i}) \quad (7)$$

$$1 = P_{uC_{oov}}^* + \sum_k P_{uC_k}^* \quad (8)$$

We eventually come to κ calculation:

$$\kappa = \left(1 + \sum_k \frac{P_{uC_k}}{1 - P_{uC_k}} \right) \prod_i (1 - P_{uC_i}) \quad (9)$$

$$\kappa = \left(1 + \sum_k L_{uC_k} \right) \prod_i (1 - P_{uC_i}) \quad (10)$$

Where $L_{uC_i} = \frac{P_{uC_i}}{1 - P_{uC_i}}$ is the likelihood of class C_i . Notice that if the applied calibration is modelled as a sigmoid with $\gamma = 0$ ¹¹, the likelihood definition is straightforward in function of x the ASR CM:

$$L(x) = e^{\alpha(x-\beta)} \quad (11)$$

¹¹The introduction of gamma was made in order to model the bias induced by the low number of classes in the yes/no question.

We eventually come to this simple formula:

$$P_{uC_k}^* = \frac{P_{uC_k} \prod_{i \neq k} (1 - P_{uC_i})}{(1 + \sum_i L_{uC_i}) \prod_i (1 - P_{uC_i})} \quad (12)$$

$$P_{uC_k}^* = \frac{L_{uC_k}}{1 + \sum_i L_{uC_i}} \quad (13)$$

The term 1 represents the C_{oov} likelihood of the input being out of vocabulary. This formula is mathematically exact and computationally simple.

A.4.2. ICE evaluation after combination

Now, the sum of probabilities cannot exceed 1, so that $P_{uC_{oov}}^*$ is always strictly positive (it equals 0 if and only if one class is affected probability 1). In order to experiment the use of this transformation, we apply it to the calibrations discussed in section A.3. This leads to the following results:

The table reveals a lot of interesting facts. The first one would be that the ASR CM barely take any advantage of combination (less than 2% reduction). On the contrary, the sigmoid optimisation combination exceeds the 26% reduction of error, which is quite impressive, considering that only one third of the corpora utterances led to a N -best with $N > 1$ and that only these ones are affected by combination. The combination based on k NN CM is halfway with a 15% reduction. Another interesting point concerns the regularity of the sigmoid evaluation over learning corpus.

The false acceptance versus false rejection visualisation in figure 2 is even more talkative. For each N -best entry, if its associated CM P (either ASR's or after sigmoid calibration and combination) is superior to a given threshold τ and the entry is incorrect, then this entry is counted as a false acceptance. Conversely, if its associated CM P is inferior to threshold τ and the entry is correct, then this entry is counted as a false rejection.

Figure 2 reveals two interesting features. The first advantage for the calibrated and combined measures is the high density of points in the high performance zone. This is the effect of the calibration. As a direct consequence to this, the value of threshold τ is easier to define for a conventional system without uncertainty management capabilities. The second advantage is the reduction of 37% of minimal false acceptance plus false rejection. This is the consequence of combination. If we did only calibration, the curve would have remained the same but with another point distribution. However, direct combination on a non-calibrated CM would show very little improvement. As a conclusion, both transformation techniques are complementary and synergistic.

Table 2: Yes/no corpora features.

Calibr. & Combin.	ASR	k NN			Sigmoid		
		MPP2	MPP3-1	MPP3-2	MPP2	MPP3-1	MPP3-2
MPP2	0.85	0.40	0.39	0.41	0.38	0.41	0.41
MPP3-1	0.89	0.52	0.44	0.50	0.41	0.40	0.40
MPP3-2	0.87	0.43	0.36	0.39	0.33	0.31	0.30
PhoDir	0.63	0.23	0.18	0.21	0.11	0.11	0.10

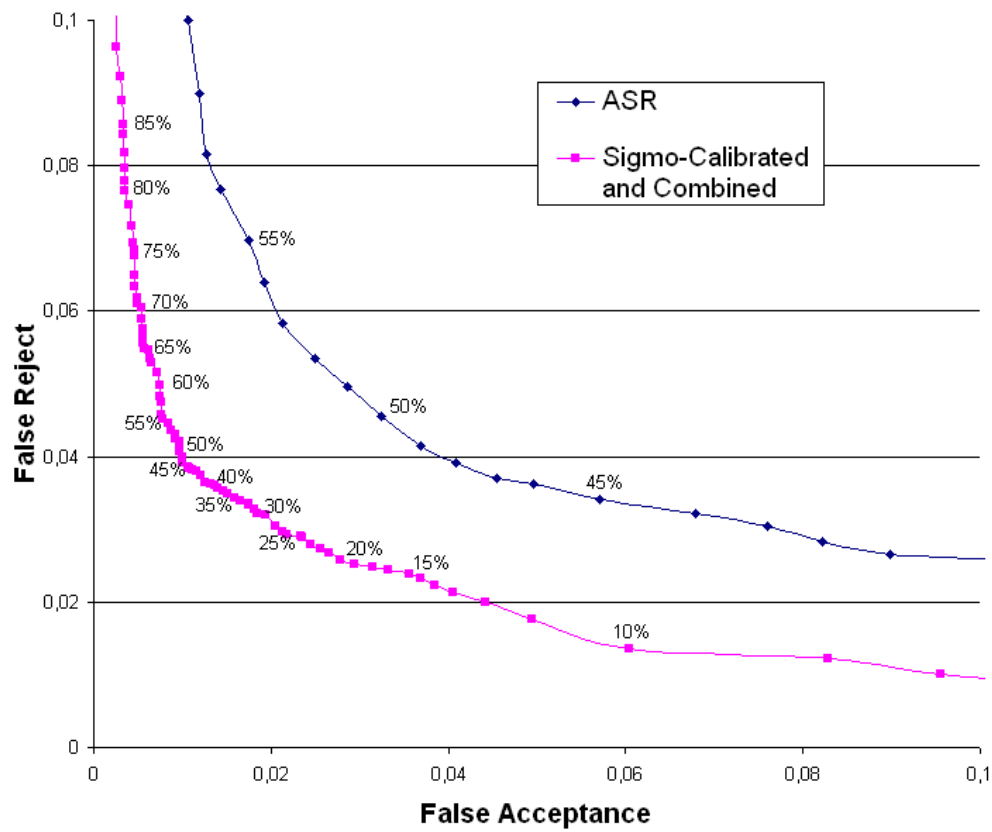


Figure 2: False acceptance versus false rejection on corpus MPP3-2 after learning with MPP2.

Figure 3 shows how the Telisma based calibrated and combined recognition outperforms the Nuance ASR result. Nuance transforms its CM into probabilities on a generic basis. The strong concentration of dots on the curve is the proof that the recognition is quite fairly calibrated. On the other hand, our method applies such a transformation on a specific learning corpus. As figures 2 and 3 conjointly illustrate, our method drastically outperforms the straightforward use of either off-the-shelf product.

A.5. Next steps

Two next steps are in progress concerning the calibration and combination algorithms in order to prove the universality of the method. First, the study of the isolated word multiclass problem: the phone directory application can retrieve a person among several thousands of names, in another application, the recognition of the town involves several dozens of thousands classes. Second, the study of the open speech multiclass problem: the commercial application on landline troubleshooting enables the customers to describe their requests naturally and then to be routed towards the relevant service.

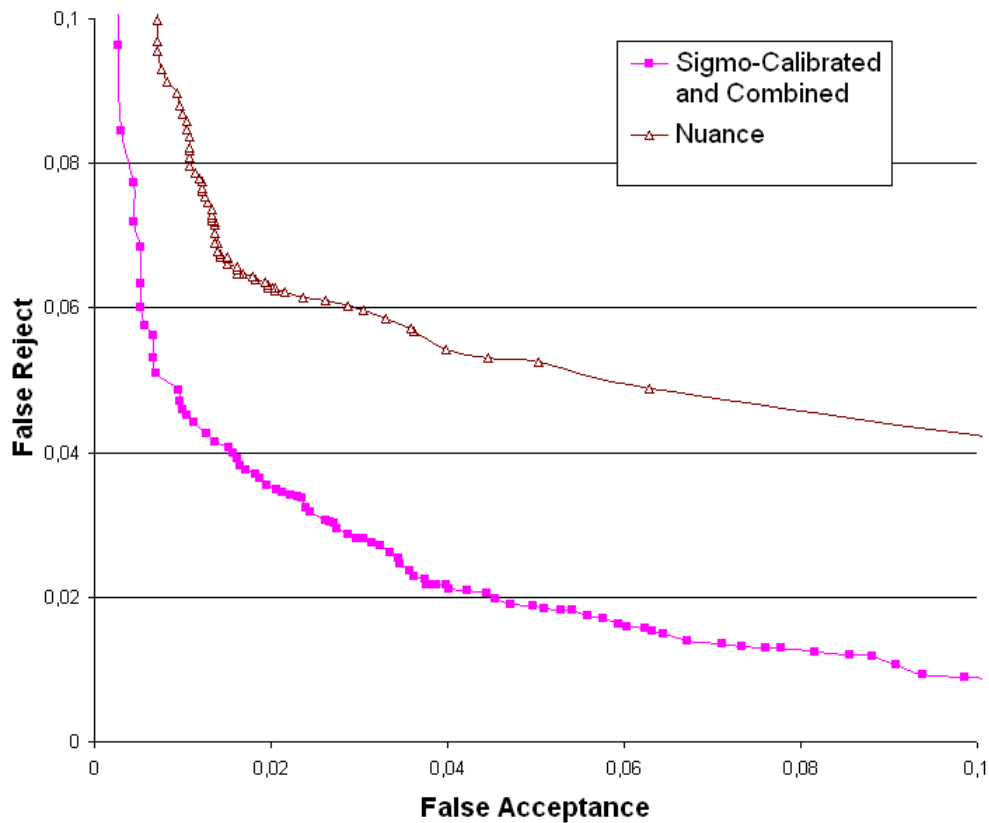


Figure 3: Comparison on the MPP2 corpus of the false acceptance vs false rejection curves between Nuance and the sigmoid calibrated on MPP3-1 and combined.

References

- [1] P. Bretier, R. Laroche, and G. Putois. Industrial self-help system (“system 3”) adapted to final architecture. Report D5.3.4, CLASSIC Project, 2010.
- [2] R. Laroche, S. Young, O. Lemon, G. Putois, and P. Bretier. Requirements analysis and theory for statistical learning approaches in automaton-based dialogue management. Report D1.1.1, CLASSIC Project, 2009.
- [3] R. Laroche, G. Putois, P. Bretier, and B. Bouchon-Meunier. Hybridisation of expertise and reinforcement learning in dialogue systems. In *Proceedings of Interspeech. Special Session: Machine Learning for Adaptivity in Spoken Dialogue*, Brighton (United Kingdom), September 2009.
- [4] G. Putois, R. Laroche, and P. Bretier. Enhanced monitoring tools and online dialogue optimisation merged into a new spoken dialogue system design experience. In *Proceedings of SIGDIAL*, Tokyo (Japan), September 2010.
- [5] R. Laroche, G. Putois, S. Young, J. Henderson, O. Lemon, V. Rieser, X. Liu, and P. Bretier. Final communication architecture and module interface definitions. Report D5.1.2, CLASSIC Project, 2010.
- [6] H. Jiang. Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4):455

– 470, 2005.

- [7] J.D. Williams and S. Balakrishnan. Estimating probability of correctness for asr n-best lists. In *Proceedings of the 10th SIGdial Conference on Discourse and Dialogue*, 2009.
- [8] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 9(3):288–298, mar 2001.
- [9] H. Jiang, F.K. Soong, and C.H. Lee. A data selection strategy for utterance verification in continuous speech recognition. In *Proceedings of Eurospeech*, pages 2573 – 2576, Aalborg (Denemark), 2001.
- [10] J.D. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 21(2):393–422, 2007.
- [11] R. Laroche, B. Bouchon-Meunier, and P. Bretier. Uncertainty management in dialogue systems. In *Proceedings of the European Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2008.
- [12] B. Thomson, K. Yu, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, and S. Young. Evaluating semantic-level confidence scores with multiple hypotheses. In *Proceedings of Interspeech*, Brisbane (Australia), August 2008.
- [13] M. Rahm and C.H. Lee. String-based minimum verification error (sb-mve) training for speech recognition. *Computer Speech & Language*, 11(2):147–160, 1997.