Dialogue Act Classification in a Spoken Dialogue System*

María José Castro[1], David Vilar[1], Pablo Aibar[2], and Emilio Sanchis[1]

[1] Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València. E-46022 València, Spain
{mcastro,dvilar,esanchis}@dsic.upv.es

[2] Departament de Llenguatges i Sistemes Informàtics
Universitat Jaume I de Castelló. E-12071 Castelló, Spain
aibar@lsi.uji.es

Abstract. A contribution to the understanding module in a spoken dialogue system is presented in this work. The task consists of answering telephone queries about timetables, prices and services for long distance trains in Spanish. In this system the representation of the meaning of an utterance is accomplished by means of *frames*, which represent the type of information of the user turn, and *cases*, which provide the information given in the sentence. The input of the understanding module is the output of the speech recognizer and its output is used by the dialogue manager.
We focus on the classification process of the dialogue user turn with respect to the second level, i.e., the identification of the type or types of *frames* given in the utterance and on the effect of the spontaneous speech recognition errors in the classification accuracy. As classifiers for the user turns we employ multilayer perceptrons, in order to use specific understanding models for each type of *frame*.

1 Introduction

Dialogue systems are one of the outstanding goals of language technology. In this kind of systems, one of the main concerns is the *understanding* of the user turns, in contrast to speech recognition systems, where the goal is the correct *transcription* of the user utterances. This allows us to ignore some words, focusing our attention on those which provide us with useful information for extracting the meaning of the utterance.

In this paper we present a contribution to the understanding module of the BASURDE [1] dialogue system. The task consists of answering telephone queries about timetables, prices and services for long distance trains in Spanish. In this system the representation of the meaning of an utterance is accomplished by means of *frames*, which represent the type of information of the user turn, and

* Thanks to the Spanish CICYT agency under contracts TIC2003-07158-C04-03 and TIC2002-04103-C03-03 for funding.

cases, which provide the information given in the sentence. The input of the understanding module is the output of the speech recognizer and its output is then used by the dialogue manager. We have tried several stochastic based approaches to the understanding module [2–5]. Recently, other approaches to specific understanding modules have been presented [6].

In this work we focus on the classification process of the dialogue user turn with respect to the second level, i.e., the identification of the type or types of frames given in the utterance and on the effect of the spontaneous speech recognition errors in the classification accuracy. Due to the multiple error sources in a dialogue system (recognition errors, unexpected answers, etc.) it is convenient to have a reliable method for detecting which type of frame has been uttered. Once the dialogue act has been determined, the posterior extraction of the attributes and values of the utterance is simplified, as we work in a restricted analysis domain. A connectionist approach to the classification problem is studied in this paper. Our previous work on this topic can be found in [7–9].

2 The dialogue structure

One of the most frequent ways to represent the dialogue structure is by using dialogue acts [10, 11], which represent the successive states of the dialogue. The labels must be specific enough to account for the different intentions of each of the turns, but general enough in order to easily adapt them to different tasks. In this work we begin with a corpus of 215 dialogues acquired using the Wizard of Oz technique, with 1 440 user turns. The reduced size of this corpus poses a problem for the correct estimation of the model parameters. With this set of dialogues we defined three levels for labeling [12]:

Dialogue Acts: This first level is task-independent and represents the general intention of the user turn. It comprises the following labels: Opening, Closing, Undefined, Not_Understood, Waiting, Consult, Acceptance, Rejection, Question, Confirmation, Answer.

***Frames*:** The second level is task-specific and represents the kind of message provided by the user, the so called *frame*. In our case we defined 15 labels which, together with their relative frequencies, are shown in Table 1.

***Cases*:** The third level takes into account the values given in the utterances, like city names, dates, etc.

The labeling of the corpus was carried out using a semiautomatic process: some dialogues were manually labeled and used to train some preliminary models, which in turn were used to label the rest of the corpus. The final result was manually reviewed. An example of the three level labeling is shown in Figure 1. One important feature of this labeling scheme is that one dialogue turn can have more than one label associated with it (see the second example of Figure 1), which allows a better specification of the meaning, but it also makes the posterior classification and segmentation tasks harder.

Table 1. The 15 frame classes and their relative frequencies

Frame Class	%
Affirmation	26.75
Departure_time	18.27
New_data	13.16
Price	12.29
Closing	10.07
Return_departure_time	5.30
Rejection	4.34
Arrival_time	3.57
Train_type	3.37
Confirmation	1.73
Not_understood	0.63
Trip_length	0.24
Return_price	0.19
Return_train_type	0.05
Return_departure_time	0.05

3 Lexicon and codification of the user turns

In the problem we are dealing with, the morphological variance of the words is not important because it does not give any additional information for the classification task. This allows us to define a set of categories and lemmas, in order to reduce the size of the vocabulary. We have defined the following categories:

1. *General categories*, like city names, week days, ordinal and cardinal numbers...
2. *Task-specific categories*, like train type or ticket type.
3. *Lemmas*: verbs in infinitive form, singular nouns and without article.

It is worth noting that some words that normally are considered "stopwords" and can be deleted in a great number of tasks, in our case they play a very important role. One clear example of this kind of words are the prepositions, that are the key to distinguish between the origin and the destination of a train. Using these prepositions we performed an additional step, splitting the general category "city_name" into the two categories "from_city_name" and "to_city_name".

After this preprocessing we reduced the size of the vocabulary from 616 to 265 words. Lastly we deleted those words with a frequency below a fixed threshold f_p (but without deleting the user turns they appear in) given that, due to their low frequency, they do not provide significant information for the discrimination of the classes. In the same way, we only considered those turns labeled with frame classes whose frequency is above the value f_c, because we do not have enough training samples available for a correct estimation of the parameters of these less frequent classes. In our case we fixed f_p and f_c both to a value of 5 (absolute frequency), which reduces the size of the corpus to 1 339 user turns, comprising a final vocabulary of 120 words and the first 10 classes of Table 1.

Original sentence:	Quería saber los horarios del Euromed Barcelona–Valencia. *I would like to know the timetables of the Euromed train from Barcelona to Valencia.*
1st level (speech act):	Question
2nd level (frames):	Departure_time
3rd level (cases):	Departure_time (Origin: barcelona, Destination: valencia, Train_type: euromed)
Original sentence:	Hola, buenos días. Me gustaría saber el precio y los horarios que hay para un billete de tren de Barcelona a La Coruña el 22 de diciembre, por favor. *Hello, good morning. I would like to know the price and timetables of a train from Barcelona to La Coruña for the 22nd of December, please.*
1st level (speech act):	Question
2nd level (frames):	Price, Departure_time
3rd level (cases):	Price (Origin: barcelona, Destination: la_coruña, Departure_time: 12/22/2002) Departure_time (Origin: barcelona, Destination: la_coruña, Departure_time: 12/22/2002)

Fig. 1. Example of the three-level labeling for two user turns. The Spanish original sentence and its English translation are given

Once the vocabulary has been fixed, the codification of each input utterance is a 120 bit vector, each bit indicating the presence or absence of a word of the vocabulary in the utterance. This coding scheme neglects the information that can be obtained taking the sequentiality of the utterance into account, but we consider that this information is not fundamental in our classification problem. This codification is a natural approach to the input format of the connectionist classifier. An example of the result of the preprocess and codification of the user turn is shown in Figure 2.

4 Multiclass classification using neural networks

We have used multilayer perceptrons (MLPs) to classify the user turns, for being one of the most widely used artificial neural networks for classification tasks. In our case the input layer gets the user turn coded as a bit vector, as explained above, and the number of output units is defined as the number of class labels of the classification task. Each unit in the (first) hidden layer defines an hyperplane in the representation space. Those hyperplanes will form the decision boundaries of the different classes. Using sigmoid activation functions, the MLPs can smooth these boundaries, adapting them to classification tasks [13]. The activation level of an output unit can be interpreted as an approximation of the a posteriori probability of the input sample belonging to the corresponding class [14].

In this way, if we face an uniclass classification problem, i.e. if the input set is formed by samples of the form

$$\{(\mathbf{x}_n, c_n)\}_{n=1}^{N}, \quad c_n \in \mathcal{C}, \tag{1}$$

Original sentence	Quería saber los horarios del Euromed Barcelona-Valencia. *I would like to know the timetables of the Euromed train from Barcelona to Valencia.*
Preprocessed sentence (“Bag of words”)	de, horario, nom_ciudad_destino, nom_ciudad_origen, querer, saber, tipo_tren *of, timetable, to_city_name, from_city_name, want, know, train_type*
▷Input local coding	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 (7 active input units)

Original sentence	Hola, buenos días. Me gustaría saber el precio y los horarios que hay para un billete de tren de Barcelona a La Coruña el 22 de diciembre, por favor. *Hello, good morning. I would like to know the price and timetables of a train from Barcelona to La Coruña for the 22nd of December, please.*
Preprocessed sentence (“Bag of words”)	buen, de, día, hola, horario, nom_ciudad_destino, nom_ciudad_origen, nom_mes, num_cardinal, querer, para, saber, tren *good, of, morning, hello, timetable, to_city_name, from_city_name, month_name, cardinal_number, want, for, know, train*
▷Input local coding	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 (13 active input units)

Fig. 2. Example of the local coding for two user turns

where $\mathcal{C} = \{c^1, c^2, \ldots, c^{|\mathcal{C}|}\}$ is the set of $|\mathcal{C}|$ class labels, the classification rule will assign the sample $\mathbf{x}$ to the class $k^\star(\mathbf{x})$ with the biggest a posteriori probability, which can be estimated using an MLP:

$$k^\star(\mathbf{x}) = \operatorname*{argmax}_{k \in \mathcal{C}} \Pr(k|\mathbf{x}) \approx \operatorname*{argmax}_{k \in \mathcal{C}} g_k(\mathbf{x}, \omega)\,, \tag{2}$$

where $g_k(\mathbf{x}, \omega)$ is the value of the kth output of the MLP given the input sample $\mathbf{x}$ and the set ω of parameters of the MLP.

In our task, however, we face a multiclass classification problem, where an utterance can have more than one dialogue act associated with it.[3] That is, the training set is constituted by samples which are pairs of the form

$$\{(\mathbf{x}_n, C_n)\}_{n=1}^{N}\,, \quad C_n \subseteq \mathcal{C}\,. \tag{3}$$

[3] In related dialogue act classification works [11] a hand segmentation of the user turns was needed in order to obtain units corresponding to one unique frame class. With this previous segmentation a n-gram based classifier was used.

Original sentence	Quería saber los horarios del Euromed Barcelona–Valencia. *I would like to know the timetables of the Euromed train from Barcelona to Valencia.*
2nd level	Departure_time
▷**Output coding**	0 1 0 0 0 0 0 0 0 0 0 (1 out of 10 classes)

Original sentence	Hola, buenos días. Me gustaría saber el precio y los horarios que hay para un billete de tren de Barcelona a La Coruña el 22 de diciembre, por favor. *Hello, good morning. I would like to know the price and timetables of a train from Barcelona to La Coruña for the 22nd of December, please.*
2nd level	Price, Departure_time
▷**Output coding**	0 1 0 1 0 0 0 0 0 0 (2 out of 10 classes)

Fig. 3. Example of the coding of the frame type or types

In our classification task, the class set $\mathcal{C}$ contains the 10 most frequent frame classes defined in Table 1. Our goal is to classify a user turn into one or more frame classes $K^\star(\mathbf{x})$ whose posterior probabilities, estimated using multilayer perceptrons, are above a threshold:

$$K^\star(\mathbf{x}) = \{k \in \mathcal{C} \mid \Pr(k|\mathbf{x}) \geq \mathcal{T}\} \approx \{k \in \mathcal{C} \mid g_k(\mathbf{x}, \omega) \geq \mathcal{T}\} \tag{4}$$

where the threshold $\mathcal{T}$ must also be estimated during the training process.

Under this approach, the classes are coded with a $|\mathcal{C}|$-dimensional bit vector, where the desired output units for each training sample are fixed to 1 for the correct frame class or classes and to 0 for the rest. Figure 3 shows an example of the coding of the desired output.

5 Experiments

For the experimentation a random splitting of the 1 339 user turns was carried out. A training set comprising about 80% of the data was formed, and the remaining 20% was used for testing. Table 2 shows the distribution of the data, along with the uniclass and multiclass frequency in each partition.

5.1 Training the MLPs

The training of the MLPs was carried out using the neural network simulation software kit "SNNS: Stuttgart Neural Network Simulator" [15]. In order to successfully use neural networks as classifiers several aspects have to be considered,

Table 2. Partition of the dataset (80% for training and 20% for test) and type of the user turns (uniclass–UC and multiclass–MC)

Data	Total	UC	MC
Training	1 071	692 (65%)	379 (35%)
Test	268	175 (65%)	93 (35%)

Table 3. MLP topologies and parameters

Topology:	One hidden layer: 2, 4, 8, 16, 32, 64 Two hidden layers: 2–2, 4–2, 4–4, 8–2, . . . , 64–64
Learning algorithm:	Backpropagation (with and without momentum term), Quickpropagation
Learning rate:	0.05, 0.1, 0.2, 0.3, 0.4, 0.5
Momentum:	0.1, 0.2, 0.3, 0.4, 0.5
Maximum increment:	1.75, 2

such as the network topology, the training algorithm and the selection of its parameters [13–15]. We carried out experiments with different network topologies, with an increasing number of units: one hidden layer with 2 units, two hidden layers with 2 units each, two hidden layers of 4 and 2 units, one hidden layer of 4 units, etc. Different learning algorithms were also used: the incremental version of the backpropagation algorithm, with and without momentum term, and the quickpropagation algorithm, studying at the same time the influence of their parameters like learning rate and momentum term. In the training process a random presentation of the samples was used. In each case a stop criterion based on a validation set was used, where a randomly chosen subset of approximately 20% of the training samples was used in order to stop the learning process and select the best configuration.

In the training phase we first tested the influence of the MLP topology. Different MLPs were trained with an increasing number of units, using the standard backpropagation algorithm, with a sigmoid activation function and learning rate equal to 0.2, selecting the best topology based on the mean squared error on the validation set.

Once the topology was fixed, we continued our experimentation training MLPs of this topology with the above mentioned algorithms, with different combinations of learning rate and momentum, and with different values of maximum increment for the quickpropagation algorithm (see Table 3).

5.2 Performance of the speech recognizer

Table 4 shows the results obtained using our speech recognizer based on semicontinous Hidden Markov Models. First, the correct transcription and the output of the recognizer are compared using the word error rate (WER) measure, and the percentage of bad recognized sentences is also given.

Secondly, the results of the same measures is given after the categorization and lemmatization explained in Section 3. The results are shown for both the training and the test partitions.

Table 4. Error rate of the speech recognition system

	Without processing		***Processed***	
Data	WER	Sentence	WER	Sentence
Training	18.71	55.56	17.53	52.90
Test	20.65	58.96	19.15	55.06

Table 5. Classification error rate of the user turns

	Test					
	Text			**Voice**		
Training	**Total**	a **UC**	**MC**	**Total**	**UC**	**MC**
Text	11.19	7.43	18.28	48.13	50.86	43.01
Text+Voice	27.24	17.71	34.70	46.64	38.29	62.37
Voice	25.00	16.57	40.86	44.40	40.57	51.61

5.3 Text and voice experiments

A common practice in understanding and voice recognition systems is to train the models with correct data (i.e., the correct transcription of the user utterances) and to test with the transcription of a speech recognizer. Connectionist classifiers, as most classifiers do, try to minimize the error of the training data and therefore a tacit assumption is made, namely that both training and test data are generated using the same model. This is not consistent with the above described approach. Therefore we have carried out a series of experiments in order to study the influence of training with text or voice data.

Learning with text data In a first phase and in order to test if the classifier can be successfully used in this task we trained an MLP (with the above described methodology) with text data. The best result on the validation set was obtained with an MLP of two hidden layers of 32 units each and training with the backpropagation algorithm with momentum term, using a learning rate equal to 0.3 and a momentum term of 0.5. In order to determine the classification threshold, the validation data was classified using values of the threshold between 0.1 and 0.9. The best classification rate was obtained using a threshold value of 0.5 (see Figure 4).

Using this MLP and the threshold equal to 0.5 we achieved an error rate[4] of 11.19% on the correctly transcribed test set. If we test this MLP (trained with text data) on the output of the speech recognizer the error rate grows up to 48.13%. All the results are shown in Table 5.

Learning with text and voice data In a second phase, starting with the above trained MLP, we retrained the classifier on the same training set, but using the data of the speech recognizer both as the training data and as the validation data for the stop criterion. With this retrained MLP an analogous process as before was carried out in order to estimate the threshold, which was fixed to a value of 0.7. The classification results, both for text and voice data, are shown in Table 5. These results show a degradation of the performance on the text data (as expected) and only a non-significant improvement on the voice data.

[4] In all the experiments, we considered a missclasification of the sample as an error. That is, in the case of the multiclass user turns we require that all the corresponding class labels are detected.

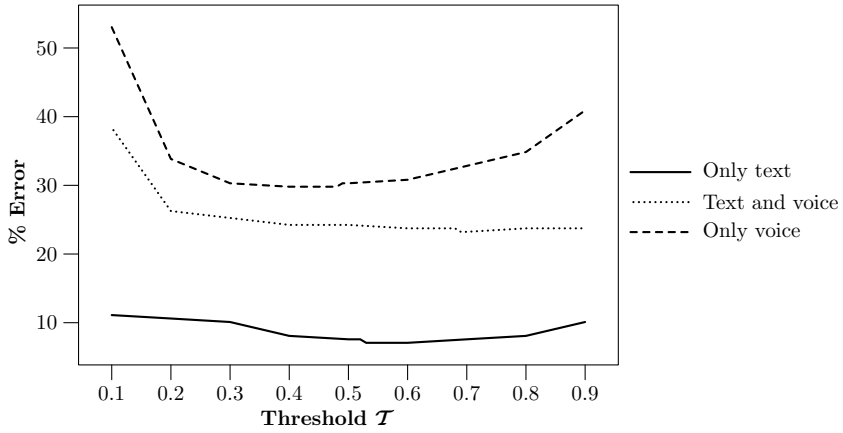

Fig. 4. Sweep on the threshold values for classification using the validation data on the three experiments

Learning with voice data Lastly, the MLPs were trained using the "real" data, i.e. directly using the data recognized by the speech recognizer system. The best topology was an MLP with two hidden layers of 16 units each. The best results were obtained using a learning rate equal to 0.2, and the best threshold value was 0.4. The results are shown in Table 5. A small improvement on the voice data is achieved, but not quite significant, perhaps due to a excessively high error rate of the speech recognizer (nearly 20% word error rate after processing the output).

6 Conclusions

The classification error rates of the user turns (Table 5) show a great degradation of performance when voice data is used. This is mostly due to the poor results of the speech recognition system (see error rate in Table 4). Nevertheless, we have empirically shown that it is convenient to use the "real" data in the training phase in order for the classifier (a MLP in our case) to be able to generalize the typical errors of the recognizer.

On the other hand, it is shown that the classification task is more difficult for multiclass user turns than for the uniclass ones. We have also tried to train and test only with the uniclass data and the results were significantly better.

Another evident conclusion of these results is that we need a two-level classification: when the confidence of the classification is high, the classification is taken as correct and the user turn will be precessed with one or more specific understanding models. If this is not the case, i.e. if the confidence of the classification is low, the turn will be rejected and a general understanding module will be used. Therefore, the estimation of the classification threshold should minimize the number of classifications errors (as we have done now) and also minimize the number of false classifications.

Lastly, we are working on the improvement of the automatic speech recognizer and the acquisition of new samples, that is, the dialogue corpus will be extended, so we will be able to repeat the experiments with a higher amount of data.

References

1. A. Bonafonte et al. Desarrollo de un sistema de diálogo oral en dominios restringidos. In *Primeras Jornadas de Tecnología del Habla*, Sevilla (Spain), 2000.
2. F. Pla, A. Molina, E. Sanchis, and F. García. Language Understanding Using Two-Level Stochastic Models with POS and Semantic Units. In *Proceedings of 4th International Conference on Text, Speech and Dialogue (TSD'01)*, 2001.
3. E. Segarra, E. Sanchis, F. García, and L. F. Hurtado. Extracting semantic information through automatic learning. In *Pattern Recognition and Image Analysis. Proceedings of IX Spanish Symposium on Pattern Recognition and Image Analysis (SNRFAI'01)*, Benicàssim (Spain), 2001.
4. Emilio Sanchis, Fernando García, Isabel Galiano, and Encarna Segarra. Applying dialogue constraints to the understanding process in a Dialogue System. In *Proc. of 5th TSD'02*, Brno (Czech Republic), 2002.
5. D. Vilar, M. J. Castro, and E. Sanchis. Connectionist classification and specific stochastic models in the understanding process of a dialogue system. In *Proc. Eurospeech'03*, Geneva (Switzerland), September 2003.
6. K. Hacioglu and W. Ward. Dialog-Context Dependent Language Modeling Combining N-grams and Stochastic Context-Free Grammars. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)*, 2001.
7. María José Castro and Emilio Sanchis. A Simple Connectionist Approach to Language Understanding in a Dialogue System. In *Advances in Artificial Intelligence – Iberamia 2002*, volume 2527 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2002.
8. Emilio Sanchis and María José Castro. Dialogue Act Connectionist Detection in a Spoken Dialogue System. In *Soft Computing Systems. Design, Management and Applications*, volume 87 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2002.
9. David Vilar, María José Castro, and Emilio Sanchis. Comparación de métodos de detección de actos de diálogo. In *Actas de las II Jornadas en Tecnologías del Habla*, Granada (España), December 2002.
10. Masaaki Nagata and Tsuyoshi Morimoto. First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 1994.
11. A. Stolcke et al. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 2000.
12. C. Martínez et al. A Labelling Proposal to Annotate Dialogues. In *Proc. LREC'02*, Las Palmas de Gran Canaria (Spain), May 2002.
13. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *PDP: Computational models of cognition and perception, I.* MIT Press, 1986.
14. C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1995.
15. A. Zell et al. *SNNS: Stuttgart Neural Network Simulator. User Manual, Version 4.2.* Institute for Parallel and Distributed High Performance Systems, University of Stuttgart, Germany, 1998.